

**SEGMENTASI PARU-PARU PADA CITRA X-RAY THORAX  
MENGUNAKAN K-MEANS**

**SKRIPSI**

oleh:

**WAHYU HARTONO**

**NIM. 12650017**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2017**

**SEGMENTASI PARU-PARU PADA CITRA X-RAY THORAX  
MENGUNAKAN K-MEANS**

**SKRIPSI**

**Diajukan kepada:**

**Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)  
Oleh:**

**WAHYU HARTONO  
NIM. 12650017**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG  
2017**

**LEMBAR PERSETUJUAN**

**SEGMENTASI PARU-PARU PADA CITRA X-RAY THORAX  
MENGUNAKAN K-MEANS**

**SKRIPSI**

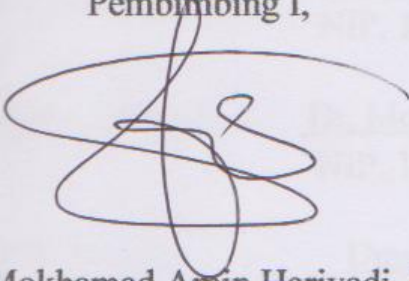
**Oleh:**

**WAHYU HARTONO**

**NIM. 12650017**

Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal: 11 September 2017

Pembimbing I,



Dr. Mokhamad Amin Hariyadi, MT  
NIP. 19670118 200501 1 001

Pembimbing II,



Umayyatus Syarifah, M. A.  
NIP. 19820925 200901 2 005

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahyo Crysdiان  
NIP. 19740424 200901 1 008

**LEMBAR PENGESAHAN**

**SEGMENTASI PARU-PARU PADA CITRA X-RAY THORAX  
MENGUNAKAN K-MEANS**

**SKRIPSI**

**Oleh :**

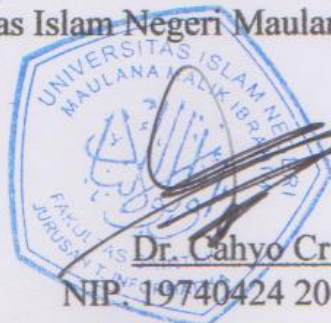
**WAHYU HARTONO**

**NIM. 12650017**

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)  
Tanggal: 30 Oktober 2017

Penguji Utama	:	<u>Dr. Cahyo Crysdian</u> NIP. 19740424 200901 1 008	(.....)
Ketua Penguji	:	<u>Irwan Budi Santoso, M.Kom</u> NIP. 19770103 201101 1 004	(.....)
Sekretaris Penguji	:	<u>Dr. Mokhamad Amin Hariyadi</u> NIP. 19670118 200501 1 001	(.....)
Anggota Penguji	:	<u>Umaiatus Syarifah, M.A</u> NIP. 19820925 200901 2 005	(.....)

Mengesahkan,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang

  
Dr. Cahyo Crysdian  
NIP. 19740424 200901 1 008

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Wahyu Hartono

NIM : 12650017

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul : Segmentasi Paru-Paru Pada Citra *X-Ray Thorax* Menggunakan *K-Means*

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada rujukan. Apabila di kemudian hari terbukti skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 25 September 2017

Yang membuat pernyataan,



Wahyu Hartono  
NIM. 12650017

## HALAMAN MOTTO

*“Sesuatu akan menjadi kebanggaan,  
Jika sesuatu itu dikerjakan,  
Dan bukan hanya dipikirkan,  
Sebuah cita-cita akan menjadi kesuksesan,  
Jika kita awali dengan bekerja untuk mencapainya,  
Bukan hanya menjadi impian”*

“kerjakanlah, wujudkanlah,  
raihlah cita-citamu  
dengan memulainya dari bekerja  
bukan hanya menjadi beban di dalam impianmu”

**“ingatlah bahwa kesuksesan selalu disertai dengan  
kegagalan”**

## HALAMAN PERSEMBAHAN

*Bismillahirrohmanirrohim, kupersembahkan sebuah karya sederhanaku ini untuk orang-orang yang paling kusayangi, kubanggakan dan selalu memberikan energi semangat untukku*

*Seluruh keluarga besarku Bani Saleh*

*Khususnya Ayah dan Ibu tercinta*

*Moch. Safi'i dan Suhartatik,*

*Yang selalu ikhlas mendoakan putra-putrinya*

*Yang selalu mengarahkan menuju jalan kebaikan*

*Yang dengan sabar membimbing selalu.*

*Semoga Allah SWT senantiasa melindungi dan menjaga mereka dalam naungannya*

*AMIN..*

## KATA PENGANTAR

*Assalamu 'alaikum Wr. Wb.*

Segala puji bagi Allah SWT tuhan semesta alam, karena atas segala rahmat dan karunia-Nya sehingga peneliti mampu menyelesaikan skripsi dengan judul “Segmentasi Paru-Paru Pada Citra *X-Ray Thorax* Menggunakan *K-Means*” dengan baik dan lancar. Shalawat serta salam selalu tercurah kepada tauladan terbaik Nabi Muhammad SAW yang telah membimbing umatnya dari zaman kebodohan menuju Islam yang *rahmatan lil alamiin*.

Dalam menyelesaikan skripsi ini, banyak pihak yang telah memberikan bantuan baik secara moril maupun materiil. Atas segala bantuan yang telah diberikan, penulis ingin menyampaikan doa dan ucapan terimakasih yang sedalam-dalamnya kepada:

1. Prof. Dr. Abdul Haris M.Ag, selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Mokhamad Amin Haryadi, MT, selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, dan mengarahkan dan memberi masukan kepada penulis dalam penulisan skripsi ini hingga akhir.
4. Umayyatus Syarifah, M. A., selaku dosen pembimbing II yang senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.
5. Umi, Abi serta keluarga besar tercinta yang selalu memberi dukungan yang tak terhingga serta doa yang senantiasa mengiringi setiap langkah penulis.



6. Dr. Cahyo Crysdiyan, selaku ketua Jurusan Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
7. Segenap Dosen Teknik Informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
8. Teman – teman seperjuangan Teknik Informatika angkatan 2012.
9. Teman – teman LP2SDM RTD dan KIM yang selalu memberikan semangat, support, doa serta motivasi penulis.

Berbagai kekurangan dan kesalahan mungkin pembaca temukan dalam penulisan skripsi ini, untuk itu penulis menerima segala kritik dan saran yang membangun dari pembaca sekalian. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya dan semoga karya ini senantiasa dapat memberi manfaat. Amim. *Wassalamualaikum Wr. Wb.*

Malang, September 2017

Penulis

## DAFTAR ISI

<b>LEMBAR PERSETUJUAN .....</b>	<b>iii</b>
<b>KATA PENGANTAR.....</b>	<b>iv</b>
<b>DAFTAR ISI.....</b>	<b>vi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah.....	5
1.5 Manfaat Penelitian.....	5
1.6 Sistematika Penulisan.....	5
<b>BAB II KAJIAN PUSTAKA .....</b>	<b>7</b>
2.1 Citra Medis .....	7
2.2 Citra <i>X-Ray</i> .....	7
2.3 <i>Thorax</i> .....	9
2.4 Paru-Paru .....	10
2.5 Segmentasi Citra <i>X-Ray</i> .....	12
2.5.1 Pre-processing Citra <i>X-Ray</i> .....	15
2.5.2 Clustering .....	16
2.5.3 Konversi Citra <i>X-Ray</i> Menjadi Citra Biner .....	21
2.5.4 Inversi Setiap <i>Bitmap</i> Citra <i>X-Ray</i> Hasil Segmentasi.....	22
2.6 Penelitian Terkait .....	23
<b>BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM.....</b>	<b>26</b>
3.1 Studi Literatur .....	26
3.2 Pengumpulan Data .....	26
3.3 Perancangan Sistem.....	27
3.4 Analisis Kebutuhan Perangkat Lunak .....	29
3.4.1 Identifikasi Pengguna Sistem .....	30
3.4.2 Daftar Kebutuhan Sistem.....	30
3.5 Sistem Segmentasi Citra <i>X-Ray</i> Menggunakan <i>K-Means</i> .....	31
3.6 Rancangan Antarmuka .....	44
3.6.1 Tab Segmentasi Data <i>X-Ray</i> .....	44
3.6.1 Tab Data Segmentasi Manual.....	46
3.6.2 Tab Pengujian .....	47
3.7 Rancangan Pengujian Sistem .....	48
<b>BAB IV IMPLEMENTASI DAN PENGUJIAN .....</b>	<b>51</b>
4.1 Spesifikasi Sistem .....	52
4.1.1 Spesifikasi Perangkat Keras .....	52
4.1.2 Spesifikasi Perangkat Lunak .....	52
4.2 Batasan Implementasi .....	53
4.3 Implementasi Segmentasi Citra <i>X-Ray</i> Menggunakan Algoritme <i>K-Means</i> .....	53
4.3.1 Pre-processing Citra <i>X-Ray</i> .....	54
4.3.2 Segmentasi Citra <i>X-Ray</i> Menggunakan Algoritme <i>K-Means</i> .....	57

4.3.3 Konversi Hasil Segmentasi Citra X-Ray Paru-Paru Menjadi Biner.....	65
4.3.4 Inversi Hasil Segmentasi Citra X-Ray Paru-Paru .....	69
4.4 Implementasi Antarmuka .....	71
4.4.1 Implementasi Tab Segmentasi Data X-Ray .....	71
4.4.2 Implementasi Tab Data Segmentasi Manual .....	72
4.4.3 Implementasi Tab Pengujian .....	73
4.5 Pengujian Dan Pembahasan .....	74
<b>BAB V PENUTUP.....</b>	<b>79</b>
5.1 Kesimpulan.....	79
5.2 Saran.....	79
<b>DAFTAR PUSTAKA .....</b>	<b>80</b>
<b>LAMPIRAN.....</b>	<b>82</b>

## ABSTRAK

Hartono, Wahyu. 2017. **Segmentasi Paru-Paru pada Citra X-Ray Thorax menggunakan K-Means**. Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Mokhamad Amin Hariyadi, MT. (II) Umairatus Syarifah, M. A.

---

**Kata kunci:** K-Means, Segmentasi, Segmentasi Thorax.

Memiliki paru-paru yang sehat merupakan salah satu hak bagi tubuh manusia. Islam mengajarkan umat manusia untuk hidup dengan sehat dan bersih sehingga akan menciptakan individu dan masyarakat yang sehat jasmani, rohani, dan sosial. Namun apabila umat mendapatkan ujian dalam bentuk sakit terutama penyakit paru-paru, masih diperlukan sebuah metode yang dapat mendeteksi penyakit berdasarkan data masukan berupa citra *x-ray thorax* dengan tingkat kesalahan yang seminimal mungkin. Oleh karena itu penulis memberikan sebuah solusi untuk menggunakan algoritme *K-Means* dengan *pre-processing* citra yang dilakukan dengan cara menghapus warna latar belakang citra *x-ray* dalam pengolahan segmentasi citra *x-ray thorax*. Sehingga akan didapatkan citra hasil segmentasi yang dapat menampilkan hanya bagian *thorax*. Pengujian *Receiver Operating Characteristics* yang telah dilakukan pada penelitian ini menghasilkan didapatkan nilai yang baik dengan nilai rata-rata akurasi sebesar 89,75%, nilai rata-rata sensitifitas sebesar 82,66%, dan nilai rata-rata spesifitas sebesar 92,90%.

## ABSTRACT

Hartono, Wahyu. 2017. **Lung Segmentation on X-Ray Thorax Images using K-Means**. Thesis. Informatics Engineering, Science and Technology Faculty, Maulana Malik Ibrahim State Islamic University of Malang. Supervisors: (I) Dr. Mokhamad Amin Hariyadi, MT. (II) Umaiatus Syarifah, M.A.

---

**Keywords:** K-Means, Segmentation, Thorax Segmentation

Having a healthy lung is one of the rights of the human body. Islam teaches mankind to live healthy and clean to create individual and society who are physically, spiritually, and socially healthy. But if the people experience the test in the form of illness, especially lung disease, they need a method that can detect the disease based on an input data in the form of x-ray image thorax with a minimum error. Therefore, the author provides a solution to use K-Means algorithm with image pre-processing performed by removing the x-ray images background on the x-ray thorax image segmentation process. Thus, it resulted the image of segmentation that can display only the thorax. Receiver Operating Characteristics test that has been done in this research resulted a good value with the average value of accuracy 89,75%, the average sensitivity value 82,66%, and the average value of specificity 92,90%.

## الملخص

هرتونو، وحي. تقطيع رئة لصور *X-Ray Thorax* باستخدام *K-Means*. بحث جامعي. شعبة الهندسة المعلوماتية، كلية العلوم والتكنولوجيا، الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. المشرفون: (أ) دوكتور محمد أمين هريدي (ب) أمية الشريفة.

الكلمات الرئيسية: *K-Means*، تقطيع، تقطيع *Thorax*.

وجود رئة صحيحة هي واحدة من حقوق لجسم الإنسان. الإسلام يعلم الناس أن يعيش بصحة جيدة ونظيفة ليحصل على الفرد والمجتمع الذين كانوا صحيحا جسديا، وروحيا، واجتماعيا. ولكن إذا كان الناس تُجرب المصيبة في شكل مرض، خاصة أمراض الرئة، فإنها لا تزال بحاجة إلى طريقة التي يمكن الكشف عن المرض على أساس بيانات المدخلات في شكل صورة الأشعة السينية الصدر مع الحد الأدنى من الخطأ. ولذلك، فإن المؤلف يوفر حلا لاستخدام خوارزمية *K-means* مع صورة معالجة مسبقة يؤديها إزالة صور الأشعة السينية الخلفية على صورة الأشعة السينية عملية تجزئة الصدر. لذلك، مما أدى إلى صورة تجزئة التي يمكن عرض الصدر فقط. اختبار خصائص التشغيل للمستقبل الذي تم القيام به في هذا البحث نتج عنه قيمة جيدة مع متوسط قيمة الدقة 89,750%، متوسط القيمة الحساسية 82,66%، ومتوسط قيمة الخصوصية 92,90%.

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Kesehatan merupakan salah satu hak bagi tubuh manusia. Dalam hal ini tujuan Islam mengajarkan hidup yang bersih dan sehat adalah menciptakan individu dan masyarakat yang sehat jasmani, rohani, dan sosial sehingga umat manusia mampu menjadi umat pilihan. Ada 2 (dua) istilah yang digunakan Islam untuk menunjuk kepada kesehatan, yaitu istilah shihhah dan ‘afiah. Dengan demikian, maka kesehatan yang dimaksud Islam adalah kesehatan fisik-biologis sekaligus kesehatan mental-psikologis.

Tidak selamanya manusia merasakan kesehatan badan yang sempurna, ada kalanya manusia dihadapkan dengan berbagai macam penyakit. Baik yang sifatnya menyerang organ luar maupun organ dalam. Berkaitan dengan hal ini, Allah SWT tidak menghendaki hamba-Nya membiarkan dirinya ketika sakit, akan tetapi diminta dan diwajibkan untuk berikhtiar dan berusaha semaksimal mungkin untuk dapat menyembuhkan penyakitnya. Secara khusus Rasulullah SAW meminta kepada sahabatnya dan umatnya untuk berobat ketika sakit, karena setiap penyakit itu pasti dapat ditemukan obatnya. Seperti dalam hadits Rasulullah SAW:

قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ أَنَّهُ : «لِكُلِّ دَاءٍ دَوَاءٌ، فَإِذَا أُصِيبَ دَوَاءُ الدَّاءِ بَرَأَ  
بِإِذْنِ اللَّهِ عَزَّ وَجَلَّ

*Artinya: " Rasulullah shallallahu 'alaihi wasallam bersabda: "Setiap penyakit ada obatnya. Apabila ditemukan obat yang tepat untuk suatu penyakit, maka akan sembuhlah penyakit itu dengan izin Allah 'azza wajalla"”.*

Berdasarkan hadits tersebut di jelaskan bahwa setiap penyakit yang di turunkan Allah SWT pasti ada obatnya. Di pihak lain, sakit dan penyakit serta resep obatnya ini menjadi tantangan tersendiri bagi para intelektual dalam bidang kedokteran untuk menemukan faktor penyebab sakitnya.

Manusia (para intelektual) khususnya dalam bidang kedokteran diberi kemampuan untuk mengetahui ciri dan hukum-hukum yang berkaitan dengan alam raya. Adanya potensi itu, menjadikan ilmuwan dapat memperoleh kepastian mengenai hukum-hukum alam. Semua itu mengantarkan manusia berpotensi untuk memanfaatkan alam yang telah Allah SWT sediakan. Keberhasilan memanfaatkan alam itu merupakan buah hasil teknologi.

Al-Quran memuji sekelompok manusia yang mengoptimalkan potensi akalinya. Ciri mereka antara lain disebutkan dalam (QS. Ali Imran/3:190-191). Berdasarkan *spirit* dari ayat tersebut, dewasa ini telah lahir teknologi khususnya pada bidang kesehatan, misalnya dalam bidang citra medis (*medical imaging*). Citra medis pada dasarnya adalah suatu teknik atau proses penggambaran bagian-bagian organ tubuh manusia dengan tujuan mengetahui kerusakan yang terdapat pada organ tubuh tersebut akibat aktivitas bakteri dan virus. Citra medis yang dimaksud di dalam penelitian ini adalah berkaitan dengan citra *x-ray thorax*. Citra *x-ray thorax* merupakan citra (gambar) yang menampilkan penampang organ tubuh bagian dalam manusia khususnya pada bagian *thorax* (rongga dada). Dalam gambar citra *x-ray thorax* terdapat bagian-bagian vital organ dalam manusia diantaranya adalah paru-paru.

Paru-paru adalah organ dalam manusia yang berfungsi menukar oksigen dari udara dengan karbondioksida dari darah dengan bantuan hemoglobin. Proses



tersebut dikenal sebagai respirasi atau pernafasan. Paru-paru terletak di dalam rongga dada (*thorax*), yang dilindungi oleh struktur tulang selangka dan diliputi dua dinding yang di kenal sebagai *pleura*. Kedua lapisan tersebut dipisahkan oleh lapisan udara yang di kenal sebagai rongga plueral yang berisi cairan plueral. (Evelyn, 2013)

Untuk mengolah hasil citra *x-ray thorax* di butuhkan proses komputerisasi agar dapat diperoleh hasil yang akurat dan akan lebih memudahkan manusia (ahli medis) untuk membaca hasil citra *x-ray thorax*. Seperti yang telah di ketahui pada kehidupan sehari-hari, dalam proses membaca hasil *x-ray* sebuah rumah sakit atau klinik masih mengandalkan dari kemampuan seorang dokter. Pada kenyataannya kemampuan seorang dokter dalam membaca hasil *x-ray* adalah bersifat unik atau kemampuan tiap dokter memiliki tingkatan akurasi yang berbeda dalam menyimpulkan diagnosa sesuai hasil membaca citra *x-ray* tersebut.

Dalam perkembangan teknologi *image processing x-ray thorax* hingga saat ini terus diperluas dengan tujuan untuk membantu manusia dalam melakukan pekerjaan. *Image processing* merupakan salah satu jenis teknologi untuk menyelesaikan masalah mengenai pemrosesan gambar, dalam *image processing x-ray thorax*, gambar yang ada diolah sedemikian rupa sehingga gambar tersebut menjadi lebih mudah untuk di proses. (Usman, 2005)

Pada proses pengolahan citra (*image processing*), dikenal sebuah istilah segmentasi. Segmentasi merupakan langkah pertama dan menjadi bagian yang penting dalam pengenalan objek (*object recognition*). Segmentasi merupakan teknik untuk membagi citra menjadi beberapa daerah (*region*) pada setiap daerah memiliki kemiripan atribut antara lain: tingkat keabuan (*grayscale*), tekstur

(*texture*), warna (*color*), dan gerakan (*motion*). Beberapa metode yang termasuk dalam segmentasi citra yaitu: (1) metode pengelompokan (*clustering method*), (2) *Histogram-Based Method*, (3) *Edge Detection Method*, (4) *Region Growing Method*, (5) *Level Set Method*, (6) *Graph Partitioning Method*, (7) *Watershed Transformation*, (8) *Model based segmentation*, (9) *Multi-scale segmentation*, *Semi-automatic segmentation*.

Dengan melihat permasalahan tersebut, penulis membangun aplikasi yang bertujuan untuk membantu kinerja para tenaga medis khususnya agar dapat membantu dalam proses membaca hasil citra *x-ray*. Salah satu metode yang dapat digunakan dalam pengolahan *image processing* adalah metode *K-Means*. Metode *K-Means* merupakan salah satu metode segmentasi yang sering digunakan dalam *image processing*.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, dapat diperoleh rumusan masalah seperti berikut:

1. Bagaimana implementasi segmentasi paru-paru pada citra *x-ray thorax* dengan menggunakan metode *K-Means*?
2. Berapa nilai akurasi yang dihasilkan dari metode *K-Means* dalam segmentasi paru-paru pada *citra x-ray thorax*?

## 1.3 Tujuan Penelitian

Dari rumusan masalah dapat diperoleh tujuan dari penelitian ini yaitu:

1. Melakukan implementasi metode *K-Means* dalam aplikasi segmentasi paru-paru pada citra *x-ray Thorax*.

2. Menghitung nilai akurasi segmentasi paru-paru pada *citra x-ray thorax* dengan menggunakan metode *K-Means*.

#### **1.4 Batasan Masalah**

Agar penyusunan penelitian ini tidak keluar dari pokok permasalahan yang dirumuskan, maka ruang lingkup pembahasan dibatasi pada:

1. Citra yang digunakan dalam penelitian hanya berupa citra *x-ray thorax* dengan warna *grayscale*.
2. Resolusi citra yang digunakan berukuran 256 x 256 dengan format *jpg (Joint Photographic Experts Group)*.

#### **1.5 Manfaat Penelitian**

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Mempermudah kinerja dokter dalam pembacaan hasil foto *x-ray thorax*, sehingga dapat mengoptimasi pengambilan keputusan.
2. Menambah wawasan mengenai ilmu kedokteran dan juga ilmu pemrograman komputer dalam bidang medis khususnya bagi peneliti.

#### **1.6 Sistematika Penulisan**

Untuk memudahkan dan memberikan gambaran yang lebih jelas dan menyeluruh mengenai penulisan laporan ini, maka dapat dilihat pada sistematika penulisan sebagai berikut.

### **BAB I Pendahuluan**

Bab pendahuluan berisi mengenai uraian latar belakang masalah, rumusan masalah, tujuan, manfaat penelitian, batasan masalah, dan sistematika penulisan laporan.

## **BAB II Landasan Teori**

Pada Bab ini terdapat landasan teori yang mencakup tentang *x-ray*, *thorax*, paru-paru, dan metode *K-Means*, serta teori-teori dari penelitian sebelumnya yang menunjang penelitian ini.

## **BAB III Perancangan Dan Implementasi Sistem**

Bab perancangan dan implementasi sistem berisi mengenai alur proses sistem di dalam aplikasi serta rancangan desain aplikasi yang dibangun pada penelitian ini.

## **BAB IV Pengujian dan Pembahasan**

Bab ini menguraikan dan membahas analisis hasil pengujian aplikasi segmentasi paru-paru pada citra *x-ray thorax* dengan menggunakan metode *K-Means*.

## **BAB V Penutup**

Bab ini berisi kesimpulan dari hasil pengujian sistem yang telah dibangun dengan saran yang diajukan oleh penulis untuk penelitian lebih lanjut

## **BAB II KAJIAN PUSTAKA**

Bab ini menjelaskan mengenai kajian dari kumpulan pustaka yang kemudian akan menjadi dasar dan landasan teori atas penelitian ini. Daftar kepustakaan yang dikaji adalah meliputi citra medis, citra *x-ray*, paru-paru, hingga proses segmentasi citra *x-ray thorax* menggunakan algoritme *K-Means*.

### **2.1 Citra Medis**

Pencitraan medis (*medical image*) adalah teknik dan proses yang digunakan untuk membuat gambar manusia secara utuh atau sebagian beserta fungsinya untuk tujuan klinis, yaitu prosedur medis yang berusaha untuk mengungkapkan keadaan anatomi dan fisiologi tubuh, mendiagnosis atau memeriksa penyakit. Sebagai disiplin dan dalam arti luas, ini adalah bagian dari pencitraan biologis dan memasukkan radiologi (dalam arti yang lebih luas) kedokteran, nuklir, investigasi, ilmu radiologis, endoskopi, (medis) *thermography*, fotografi medis dan mikroskopi (misalnya untuk penyelidikan patologis manusia). Pengukuran dan teknik perekaman yang dirancang untuk menghasilkan gambar, seperti *electroencephalography* (EEG), *magnetoencephalography* (MEG), *electrocardiography* (EEG) dan lain sebagainya, tetapi yang menghasilkan data yang rentan untuk diwakili sebagai peta (yaitu yang berisi informasi posisi), dapat dilihat sebagai bentuk pencitraan medis (Sabar, 2009).

### **2.2 Citra X-Ray**

Wilhelm Conrad Rontgen adalah seorang ahli fisika dari Universitas Wurzburg, Jerman, yang pertama kali menemukan sinar rontgen pada tahun 1895 sewaktu melakukan percobaan dengan sinar katoda. Pada saat itu beliau melihat

timbulnya sinar flourensensi yang berasal dari Kristal barium platisianida dalam tabung crookes hittorf yang dialiri listrik. Beliau segera menyadari bahwa fenomena ini merupakan suatu penemuan baru sehingga dengan gigih beliau terus menerus melanjutkan penyelidikannya dalam minggu-minggu berikutnya. Tidak lama kemudian ditemukanlah sinar yang disebutnya sinar sinar baru atau *x-ray*, dikemudian hari masyarakat menamakan sinar tersebut sebagai sinar *rontgen* sebagai penghargaan kepada Wilhelm Conrad Rontgen. (Gabriel, 1996 & Ainatul, 2011)

*X-ray* merupakan suatu bentuk radiasi seperti cahaya atau gelombang bunyi. *X-ray* dapat melewati banyak objek termasuk tubuh. Mesin *x-ray* menghasilkan pancaran radiasi kecil yang melewati tubuh, merekam citra pada film atau plat khusus untuk merekam gambar digital. Masing-masing bagian tubuh menyerap *x-ray* dengan dosis bervariasi. Tulang yang padat menerima radiasi yang lebih besar dibanding bagian yang lebih lunak seperti otot, lemak ataupun organ tubuh. Sebagian hasilnya, tulang tergambar putih pada *x-ray*, bagian yang lunak tergambar abu-abu dan udara hitam. Pada *x-ray* dada, tulang rusuk dan tulang belakang akan menyerap banyak radiasi dan tampak abu-abu putih atau terang pada gambar. Jaringan paru-paru menyerap radiasi kecil dan akan terlihat gelap pada gambar.

Pada sistem pencitraan *x-ray* diperlukan tegangan tinggi, dengan tujuan agar dapat dihasilkan berkas *x-ray*. Untuk itu rangkaian listriknya dirancang sedemikian rupa sehingga tegangan tinggi (*kV*) dengan rentang yang besar. Jika *kV*-nya rendah maka sinar-*x* memiliki gelombang yang panjang sehingga akan mudah diserap oleh atom dari target (anoda), kemudian disebut sebagai soft *x-ray*. Radiasi yang dihasilkan dengan pengaturan tegangan yang cukup tinggi maka akan dihasilkan *x-*

ray dengan daya tembus yang besar dan panjang gelombang yang pendek (Ferry Suyanto, 2008).

Manfaat x-ray dalam ilmu kedokteran, yaitu x-ray dapat digunakan untuk melihat kondisi tulang, gigi, paru-paru serta organ tubuh yang lain tanpa melakukan pembedahan langsung pada tubuh pasien. Selain bermanfaat, x-ray mempunyai efek atau dampak yang sangat berbahaya bagi tubuh kita yaitu apabila digunakan berlebihan akan menimbulkan penyakit yang berbahaya, seperti kanker. Oleh sebab itu, para dokter tidak menganjurkan terlalu sering memakai “foto *rontgen*” secara berlebihan (Gabriel, 1996).



**Gambar 2.1** Gambar citra x-ray thorax

Sumber: <http://www.isi.uu.nl/Research/Databases/SCR/>

### **2.3 Thorax**

*Thorax* (rongga dada) adalah daerah tubuh yang terletak diantara leher dan *abdomen*. *Thorax* rata dibagian depan dan belakang tetapi melengkung di bagian samping. Rangka dinding *thorax* yang dinamakan *cavea thoracis* dibentuk oleh bagian *columna vertebralis* dibelakang, *costae* dan *spatium intercostale* disamping, serta *sternum* dan *cartilage costalis* didepan. Dibagian atas, *thorax* berhubungan dengan leher dan dibagian bawah dipisahkan dari *abdomen* oleh *diaphragma*.

*Cavea thoracis* melindungi paru dan jantung dan merupakan tempat perlekatan otot-otot *thorax*, *extremitas superior*, *abdomen* dan punggung.

*Cavitas thoracis* (rongga *thorax*) dapat dibagi menjadi: bagian tengah yang disebut *mediastinum* dan bagian lateral yang ditempati *pleura* dan paru. Paru diliputi oleh selapis *membrane* tipis yang disebut *pleura viscelaris*, yang beralih di hilus *pulmonalis* (tempat saluran udara utama dan pembuluh darah masuk ke paru-paru) menjadi *pleura parietalis* dan menuju ke permukaan dalam dinding *thorax*. Dengan cara ini terbentuk dua kantong *membranosa* yang dinamakan *cavitas pleuralis* pada setiap sisi *thorax*, diantara paru-paru dan dinding *thorax* (Richard S. Snell, 2006).

#### **2.4 Paru-Paru**

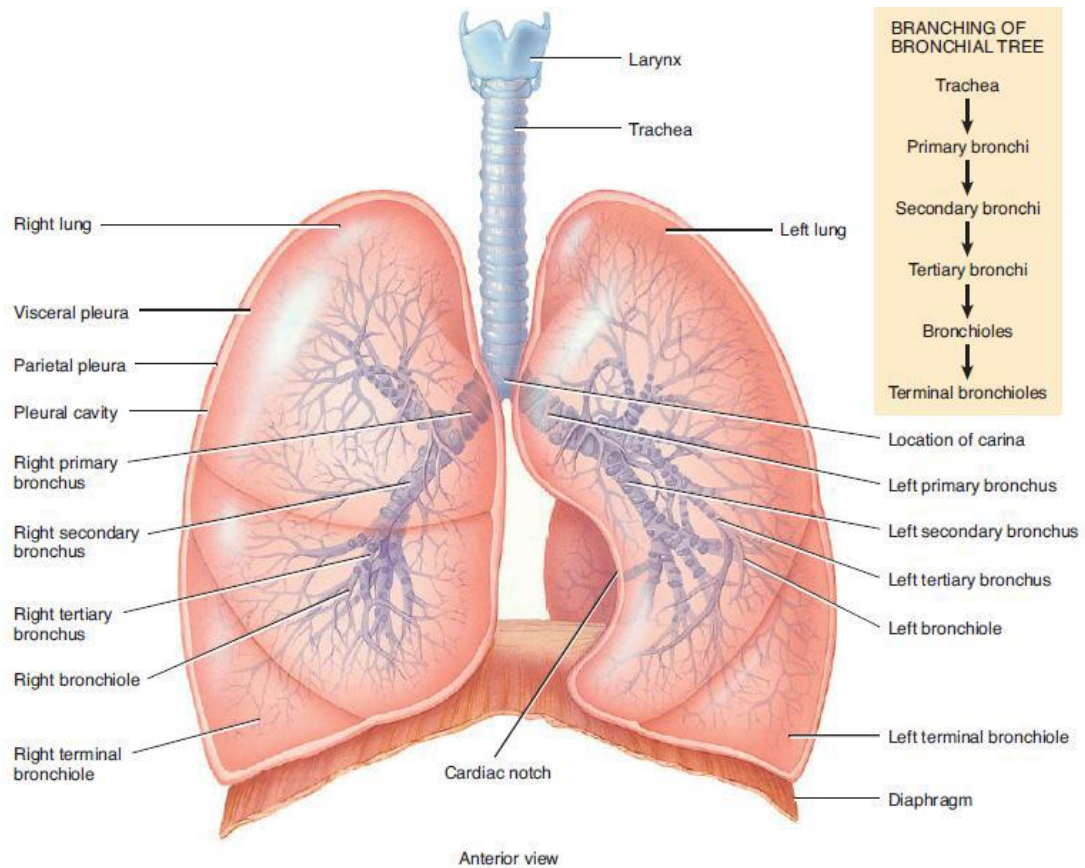
Paru-paru merupakan salah satu organ vital pada tubuh manusia yang memiliki fungsi utama sebagai alat respirasi di dalam tubuh manusia, paru-paru secara spesifik memiliki peran dalam proses pertukaran oksigen ( $O_2$ ) dengan karbon dioksida ( $CO_2$ ). Pertukaran ini terjadi pada alveolus di paru-paru melalui sistem kapiler. Paru-paru terletak pada rongga dada. Bentuk paru-paru yaitu kerucut yang ujungnya berada di atas tulang iga pertama dan dasarnya berada pada diafragma. Paru terbagi menjadi dua yaitu, paru kanan dan paru kiri. Paru-paru kanan mempunyai tiga lobus sedangkan paru-paru kiri mempunyai dua lobus. Pada paru-paru kanan terdapat tiga lobus antara lain yakni lobus superior, lobus medius dan lobus inferior. Sementara pada paru kiri hanya terdapat lobus superior dan lobus inferior. Pada paru kiri terdapat satu bagian di lobus superior yang analog dengan lobus medius paru kanan, yakni disebut sebagai *lingula pulmonis*. Diantara lobus-lobus paru kanan terdapat dua fissura, yakni *fissura horizontalis* dan *fissura*



*obliqua*, sementara di antara *lobus superior* dan *lobus inferior* paru kiri terdapat *fissura obliqua*. Setiap paru-paru terbagi lagi menjadi beberapa sub bagian menjadi sekitar sepuluh unit terkecil yang disebut *bronchopulmonary segments*. Paru-paru kanan dan paru-paru kiri dipisahkan oleh ruang yang disebut dengan *mediastinum* (Sherwood, 2001).

Paru-paru sendiri memiliki kemampuan *recoil*, yakni kemampuan untuk mengembang dan mengempis dengan sendirinya. Elastisitas paru-paru untuk mengembang dan mengempis ini disebabkan karena adanya *surfactant* yang dihasilkan oleh sel *alveolar*. Namun selain itu, mengembang dan mengempisnya paru-paru juga karena dibantu oleh otot-otot dinding *thorax* dan otot pernafasan lainnya, serta tekanan negative yang terdapat didalam *cavum pleura*.

Paru manusia terbentuk setelah embrio mempunyai panjang 3 mm. Pembentukan paru-paru di mulai dari sebuah *Groove* yang berasal dari *foregut*. Pada *groove* terbentuk dua kantung yang dilapisi oleh suatu jaringan yang disebut *primary lung bud*. Bagian proksimal *foregut* membagi diri menjadi 2 yaitu esofagus dan trakea. Pada perkembangan selanjutnya trakea akan bergabung dengan *primary lung bud*. *Primary lung bud* merupakan cikal bakal bronchi dan cabang-cabangnya. *Bronchial-tree* terbentuk setelah embrio berumur 16 minggu, sedangkan *alveoli* baru berkembang setelah bayi lahir dan jumlahnya terus meningkat hingga anak berumur 8 tahun. *Alveoli* bertambah besar sesuai dengan perkembangan dinding toraks. Jadi, pertumbuhan dan perkembangan paru berjalan terus menerus tanpa terputus sampai pertumbuhan *somatic* berhenti (Evelyn, 2009).



**Gambar 2.2** Anatomi Paru-paru

Sumber: Tortora (2012)

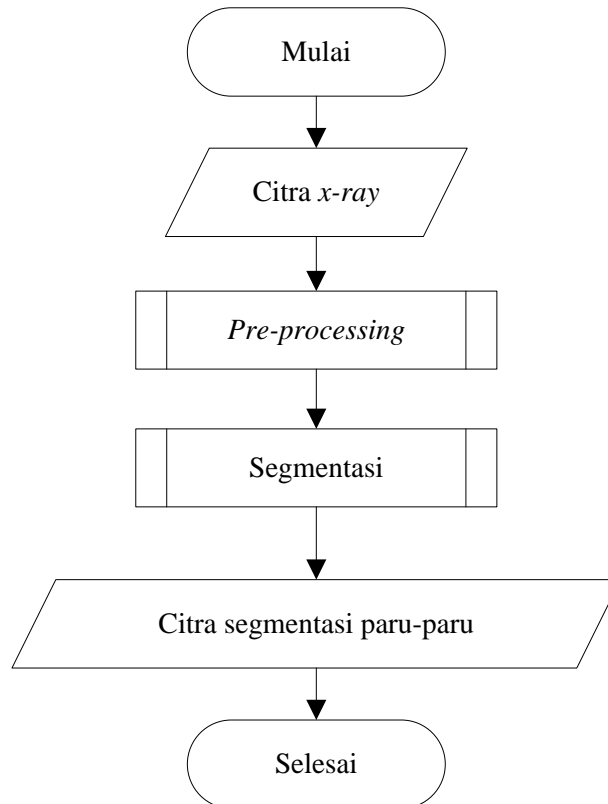
### 2.5 Segmentasi Citra *X-Ray*

Untuk melakukan manipulasi pada suatu objek dalam citra tentunya bukan suatu hal yang mudah. Akan menjadi sulit apabila melakukan manipulasi suatu citra tanpa menyentuh objek lainnya, karena objek tersebut masih bercampur dengan objek-objek lain. Sehingga untuk dapat memisahkan objek tersebut diperlukan suatu metode pengolahan citra, sehingga dapat diperoleh objek citra yang diinginkan. Salah satu metode agar dapat memisahkan atau memfokuskan pada objek yang akan diinginkan adalah dengan menggunakan *segmentasi* citra.

Segmentasi citra bertujuan untuk membagi wilayah-wilayah yang homogen. Segmentasi adalah salah satu metode penting yang digunakan untuk mengubah citra

input ke dalam citra output berdasarkan atribut yang diambil dari citra tersebut. Segmentasi membagi citra ke dalam daerah intensitasnya masing-masing sehingga bisa membedakan antara objek dan *background*-nya. Pembagian ini tergantung pada masalah yang akan diselesaikan. Segmentasi harus dihentikan apabila masing-masing objek telah terisolasi atau terlihat dengan jelas. Tingkat akurasi segmentasi bergantung pada tingkat keberhasilan prosedur analisis yang dilakukan. Dari proses tersebut diharapkan proses segmentasi memiliki tingkat keakuratan yang tinggi (Sutoyo, 2009).

Proses segmentasi citra *x-ray* terdiri dari dua tahapan sehingga dihasilkan segmentasi citra akhir yang merupakan hasil dari sistem yang dibangun pada penelitian ini. Data nilai masukan berdasarkan citra yang digunakan adalah nilai  $\alpha$ RGB ( $\alpha$ , R, G, B). terdapat 4 data nilai yang terkandung di dalam nya. Nilai  $\alpha$  merupakan nilai transparansi, dimana bila suatu *pixel* bernilai 0% pada nilai  $\alpha$ -nya, berarti citra tersebut ditampilkan benar-benar tranparan (*invisible*), sedangkan sebaliknya apabila suatu *pixel* bernilai 100% pada nilai  $\alpha$ -nya, berarti citra tersebut ditampilkan tidak transparan (*traditional digital images*). Nilai R berarti warna merah, nlai G berarti hijau, dan nilai B berarti biru, kompilasi nilai ketiga nilai RGB akan menampilkan sebuah warna yang ditampilkan pada suatu *pixel* pada sebuah citra.



**Gambar 2.3** Diagram Alir Segmentasi Citra *X-Ray* Paru-Paru

Berdasarkan Gambar 2.3, Proses segmentasi citra *x-ray* yang digunakan sistem dijabarkan kembali berikut:

1. Pre-processing citra *x-ray*
  - a. *Filtering* warna latar belakang.
2. Segmentasi citra *x-ray*
  - a. Klustering warna citra dengan menggunakan algoritme *K-Means*.
  - b. Konversi citra hasil segmentasi *x-ray* menjadi citra biner.
  - c. Inversi setiap *bitmap* citra segmentasi *x-ray* dari putih menjadi hitam.

### 2.5.1 Pre-processing Citra X-Ray

*Pre-prosesing* citra merupakan proses pengolahan awal data-data citra untuk dianalisis lebih lanjut. *Pre-procesing* dapat melakukan baik pembersihan *noise* pada citra, pengubahan *format* warna citra, maupun proses deteksi *edge* atau bagian tepi pada citra. Beberapa proses yang ada diantaranya adalah komposit, *cropping* dan mozaik citra. Adapun tahapan-tahapan *pre-prosesing* pada penelitian ini adalah sebagai berikut:

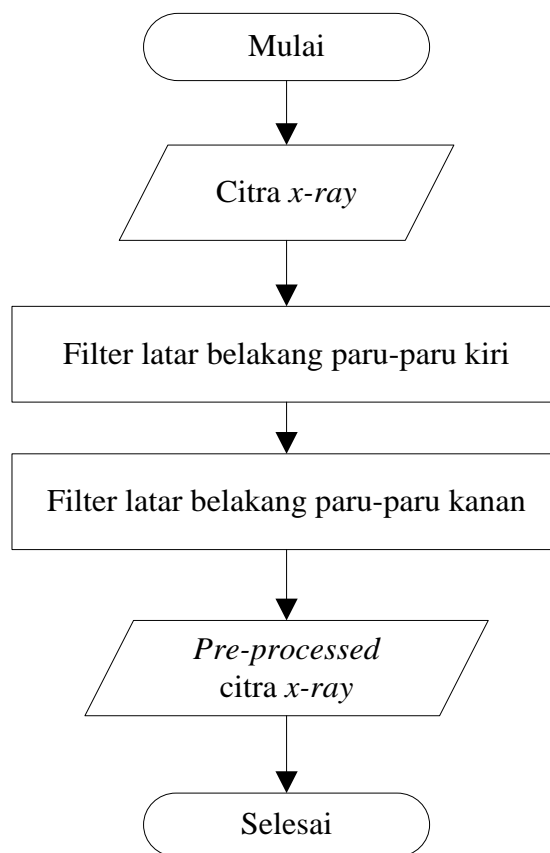
#### 2.5.1.1 Menghapus Warna Latar Belakang

Proses *pre-processing* yang digunakan pada penelitian ini dilakukan dengan melakukan deteksi warna latar belakang citra *x-ray*. Deteksi warna latar belakang dilakukan dengan melakukan pengubahan warna hitam (*RGB*: 0, 0, 0) dengan warna putih (*RGB*: 255, 255, 255).

Proses deteksi warna latar belakang diawali dengan cara mengubah skala gambar dari pixel menjadi *bitmap* dengan resolusi 256 x 256 sehingga didapatkan panjang resolusi *bitmap* sepanjang 256-bit dan lebar sepanjang 256-bit. Karena data masukan citra sudah berupa *grayscale*, maka tidak perlu dilakukan konversi citra menjadi *grayscale*, sehingga dapat langsung dilakukan *scanning* setiap *bitmap* dari data *input* citra. Proses *scanning* dilakukan pada paru-paru kanan dan paru-paru kiri secara terpisah dengan menggunakan fungsi perulangan (*looping*).

Proses *scanning loop* dilakukan dengan meletakkan pada koordinat *bitmap* (0, 0) untuk paru-paru kiri dan koordinat *bitmap* (0, 255) untuk paru-paru kanan. Kemudian pada koordinat *bitmap* tersebut dilakukan *filter* apakah warna *bitmap* yang ada pada koordinat tersebut kurang dari *RGB* (128, 128, 128) menjadi warna putih dengan *RGB* (255, 255, 255). Nilai *bitmap* (128, 128, 128) adalah nilai batas

atas untuk warna hitam. Batas nilai ini digunakan karena pada data *input* citra *x-ray*, bagian thorax memiliki nilai bitmap yang berkisar pada warna abu-abu agak terang hingga warna putih dengan nilai  $RGB > (128, 128, 128)$ . Sedangkan latar belakang citra memiliki warna hitam pekat dengan kisaran nilai  $RGB < (128, 128, 128)$ . Langkah proses pre-processing citra *x-ray* dilakukan untuk paru-paru kiri dan paru-paru kanan ditampilkan pada Gambar 2.4.



**Gambar 2.4** Diagram Alir *Filter* Latar Belakang Citra *X-Ray* Paru-Paru

### 2.5.2 Clustering

*Clustering* atau yang sering dikenal dengan istilah klastering dalam Bahasa Indonesia adalah proses pengelompokan objek yang memaparkan hubungan dari setiap objek dengan menggunakan prinsip memaksimalkan nilai yang sama untuk setiap anggota yang berada di dalam satu klaster, dan meminimumkan nilai yang

sama untuk setiap klaster yang berada di dalam satu klaster. Tujuan utama dari klastering adalah untuk mendapatkan klaster yang berkualitas dengan efisien dari segi waktu eksekusinya. Klastering di dalam *data mining* berfungsi untuk mendapatkan pola dari distribusi yang ada di dalam sekumpulan set data yang digunakan untuk proses analisis data. Jarak kedekatan setiap nilai pada masing-masing atribut adalah parameter yang digunakan untuk menentukan sama atau tidaknya setiap objek di dalam satu klaster, sedangkan sebuah titik pusat (*centroid*) yang ada di dalam ruang multidimensi set data merupakan nilai representasi dari sekumpulan objek set data. Kategori algoritme klastering yang dikenal secara gasir besar ada dua, yaitu:

1. Metode Partisi

Metode partisi diawali dengan penentuan jumlah partisi  $k$  oleh *user*. Set data yang dimasukkan pada salah satu partisi telah diuji sebelumnya agar tidak terjadi *overlap*. Contoh metode partisi: algoritme *K-Means*.

2. Metode Hirarki

Metode hirarki digunakan untuk menghasilkan suatu data yang memiliki lebih dari satu klaster, di mana klaster yang satu bisa jadi merupakan turunan dari klaster yang lain. Hal ini dapat diistilahkan dengan klaster bersarang (*nested cluster*). Prinsip dari metode hirarki terbagi menjadi dua, yaitu *bottom-up* dan *top-down*. *Top-down* adalah klaster utama yang kemudian dipecah menjadi klaster yang lebih kecil ke *level* di bawahnya, sedangkan *bottom-down* adalah sebaliknya. Contohnya metode hirarki: *agglomerative (FINDIT, PROCLUS)* dan *divisive hierarcycal clustering (CLIQUE, MAFIA, dan ENCLUE)*.

### 2.5.2.1 Algoritme *K-Means*

Algoritme *K-Means* merupakan salah satu teknik klastering yang paling umum dan sederhana. Algoritme *K-Means* dilakukan dengan cara pengelompokan objek ke dalam  $k$  klaster/ kelompok. Nilai  $k$  harus ditentukan terlebih dahulu (berbeda dengan *hierarchical clustering*). Ukuran ketidakmiripan masih tetap digunakan untuk mengelompokkan objek yang ada. Algoritme ini digunakan pada penelitian ini secara khusus untuk menangani klasterisasi atau pengelompokan warna pada citra *x-ray* pada *thorax* supaya mendapatkan hasil citra yang hanya memiliki dua jenis warna sehingga citra tersebut akan lebih mudah untuk dilakukan proses lebih lanjut.

Secara ringkas proses algoritme *K-Means* adalah sebagai berikut:

1. Tentukan jumlah klaster  $k$ .
2. Inisialisasi penentuan pusat klaster (*centroid*)  $k$ .
3. Tempatkan setiap data/ objek pada klaster terdekat.
4. Perhitungan kembali penentuan pusat klaster (*centroid*) baru dengan cara menghitung nilai rata-rata dari data yang ada pada pusat klaster yang sama.
5. Ulangi langkah ke-3 dan langkah ke-4 dengan memakai pusat klaster yang baru. Jika pusat klaster tidak berubah lagi dan kondisi konvergen tercapai, maka proses klasterisasi dihentikan. Kondisi konvergen adalah perubahan fungsi objektif sudah di bawah ambang batas yang diinginkan, atau tidak ada data yang berpindah klaster. Sehingga sudah tidak ada data yang berpindah klaster dan posisi pusat klaster sudah di bawah ambang batas yang sudah ditentukan.



Perhitungan jarak antara setiap data dan pusat kluster dilakukan dengan menggunakan persamaan *Euclidean Distance*, sesuai dengan Persamaan 2.1.

$$d_{ij} = \sqrt{\sum_{k=2}^p (x_{ik} - x_{jk})^2} \dots\dots\dots(2.1)$$

Keterangan:

$d_{ij}$  = Jarak objek antara objek  $i$  dan  $j$

$P$  = Dimensi data

$x_{ik}$  = Kordinat dari objek  $i$  pada dimensi  $k$

$x_{jk}$  = Kordinat dari objek  $j$  pada dimensi  $k$

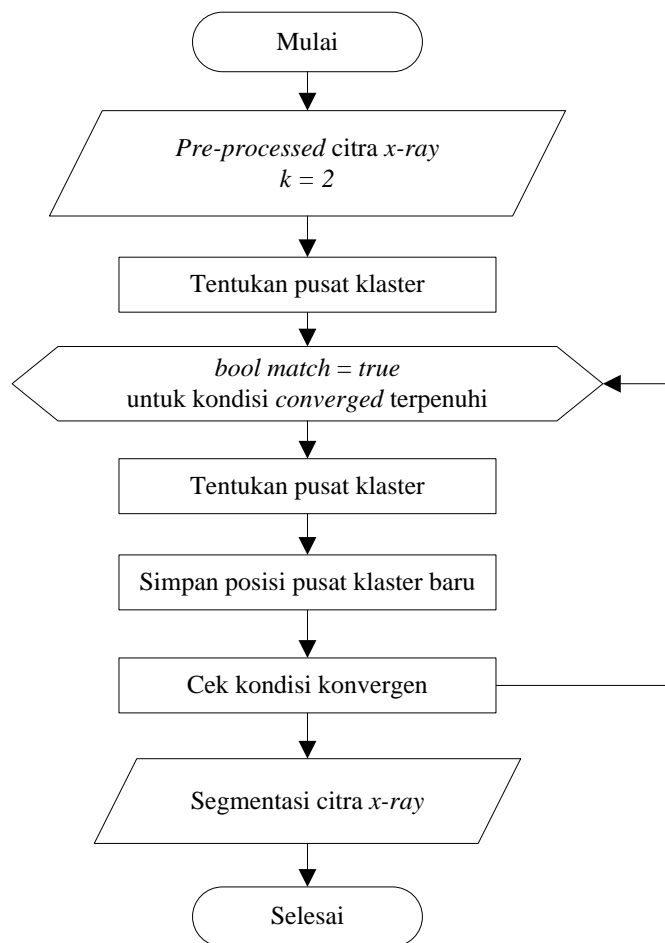
Penjabaran langkah-langkah klustering dengan menggunakan algoritme *K-Means* untuk implementasi dalam penelitian ini adalah sebagai berikut:

- a. Sistem menerima nilai *input k*. Kemudian menentukan pusat kluster dari kordinat (0, 0).
- b. Kelompokkan setiap data ke dalam kluster dengan jarak terdekatnya.
- c. Update posisi kordinat pusat kluster yang baru.
- d. Melakukan cek kondisi konvergen. Apabila kondisi konvergen terpenuhi, maka iterasi pencarian pusat kluster yang baru dihentikan, apabila tidak, maka akan sistem akan melakukan iterasi pencarian pusat kluster yang baru.
- e. *Output* dari sistem adalah sebuah citra yang hanya memiliki dua warna saja sehingga batas warna sudah tampak dengan jelas.

Pada Gambar 2.5, proses penentuan pusat kluster adalah penentuan kordinat pusat untuk kluster (*centroid*). Penentuan kordinat pusat dari citra yang pertama adalah dengan mencari dua warna yang memiliki nilai *pixel* paling tinggi. Dalam

hal ini adalah warna putih ( $RGB: 255, 255, 255$ ) dan warna hitam ( $RGB: 0, 0, 0$ ). Kemudian dua titik *pixel* yang memiliki dua warna tersebut dijadikan sebagai pusat klaster yang pertama. Setelah didapatkan dua pusat klaster tersebut, berikutnya dilakukan pembacaan warna ( $RGB$ ) setiap *pixel* pada citra dan mencari jaraknya kepada setiap 2 pusat klaster tersebut. Pencarian jarak setiap *pixel* ke pusat klaster digunakan Persamaan 2.1 *Euclidean Distance* dengan menggunakan model citra  $RGB$ . Setelah didapatkan jarak masing-masing *pixel*, kemudian nilai *pixel* tersebut dipindahkan ke pusat klaster terdekatnya. Setelah pemindahan *pixel* ke klaster terdekatnya masing-masing, ditentukan dua klaster baru, dengan cara mencari kembali dua warna dengan nilai *pixel* yang paling besar, kemudian dibandingkan nilai  $RGB$  dari dua klaster baru dengan nilai  $RGB$  dari dua klaster yang sebelumnya. Nilai klaster baru didapatkan dengan menghitung rata2 warna dari citra. Kondisi konvergen ditentukan untuk melihat apakah dua pusat klaster yang baru sama dengan dua pusat klaster sebelumnya atau tidak, apabila sama, maka akan ditentukan pusat klaster kembali dan dilakukan pengelompokkan nilai *bitmap* nya kembali. Namun apabila pusat klaster tidak berubah dari sebelumnya, berarti dua pusat klaster yang baru telah stabil.

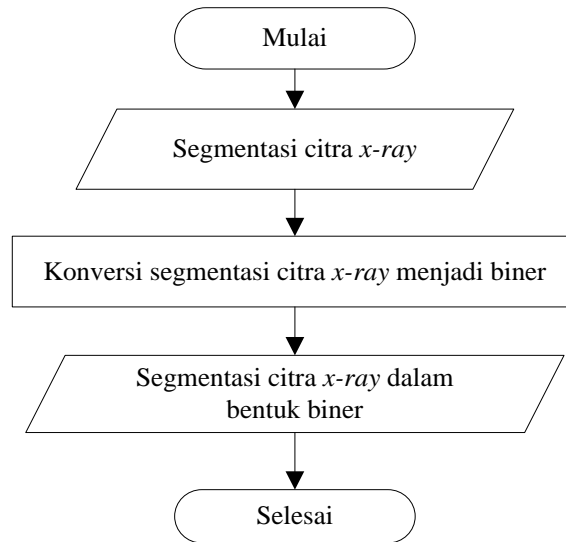
Proses dari klastering data citra *x-ray* dengan menggunakan metode *K-Means* ditampilkan pada Gambar 2.5.



**Gambar 2.5** Diagram Alir Algoritme *K-Means*

### 2.5.3 Konversi Citra X-Ray Menjadi Citra Biner

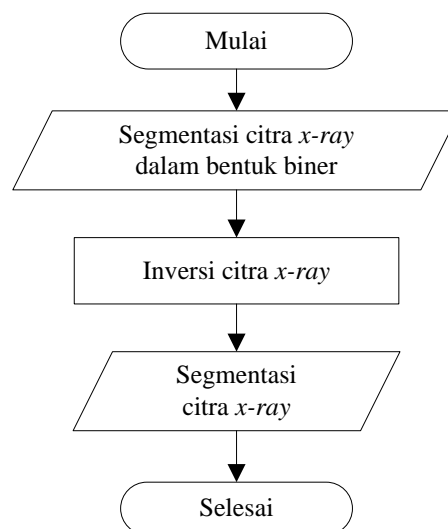
Konversi citra *x-ray* dilakukan setelah didapatkan citra hasil segmentasi dengan algoritme *K-Means*. Konversi citra dilakukan dengan cara mengambil dua nilai warna *bitmap* yang paling gelap yang telah dihasilkan dari proses segmentasi algoritme *K-Means*. Kemudian dilakukan *scanning* untuk setiap *bitmap* pada citra hasil segmentasi dan mengubah nilai *pixel* warnanya. Apabila warna pada setiap kordinat *bitmap* adalah yang paling gelap, maka diganti menjadi warna hitam pekat (*RGB*: 0, 0, 0) dan begitu juga sebaliknya seperti yang ditampilkan pada Gambar 2.6.



**Gambar 2.6** Diagram Alir Konversi Hasil Segmentasi Citra Menjadi Biner

#### 2.5.4 Inversi Setiap *Bitmap* Citra X-Ray Hasil Segmentasi

Inversi *bitmap* dilakukan untuk mengubah bagian warna pada citra *x-ray* hasil dari proses segmentasi. Bagian warna putih diubah menjadi warna hitam, dan warna hitam diubah menjadi putih. Hal ini dikarenakan pada pencocokan dan pengujian dengan melakukan perbandingan pada citra hasil segmentasi secara manual, bagian *thorax* adalah berwarna putih, dan daerah di sekitar *thorax* berwarna hitam. Proses inversi citra *bitmap* ditampilkan pada Gambar 2.7.



**Gambar 2.7** Diagram Alir Inversi *Bitmap* Citra X-Ray Hasil Segmentasi

## 2.6 Penelitian Terkait

Sebuah penelitian dilakukan oleh Amin Hariyadi, Linda Salma Angraeni, dan A. Syahirul Fatkhurrahman (2015), Teknik Informatika Universitas Islam Negeri Mulana Malik Ibrahim Malang yang berjudul “Aplikasi Segmentasi Paru-Paru”. Pada penelitian ini, telah diimplementasikan sebuah perangkat lunak segmentasi citra *x-ray thorax* dengan menggunakan metode *Chan-veese* digunakan pada proses segmentasinya. Berdasarkan hasil pengujian menggunakan metode *Mean Square Erro* yang yang telah dilakukan, didapatkan rata-rata nilai akurasi sebesar 87,89% sensitifitas sebesar 76,27%, dan spesifitas sebesar 93,97%, dengan nilai akurasi terbesar adalah 95,9592 dan nilai akurasi terendah sebesar 82,0419. Nilai sensitifitas tertinggi adalah sebesar 83,1183 dan nilai sensitifitas terendah sebesar 68,3135. Sedangkan nilai spesifitas tertinggi sebesar 98,706 dan nilai spesifitas terendah adalah sebesar 88,3848. Kajian dari penelitian tersebut digunakan untuk dilakukan perbandingan hasil pengujian dari penelitian ini, di mana pada penelitian tersebut digunakan metode *Chan-Vese*, sedangkan pada penelitian ini digunakan metode *K-Means*.

Sebuah penelitian tentang segmentasi citra *x-ray* paru-paru telah dilakukan oleh Ainatul Mardhiyah dan Agus Harjoko (2011), Universitas Islam Negeri Maulana Malik Ibrahim Malang yang berjudul “*Metode Segmentasi Paru-Paru dan Jantung Pada Citra X-Ray Thorax*”. Pada penelitian tersebut diterapkan metode *Gaussian lowpass Filter* pada tahap *pre-processing*-nya. Pada tahap segmentasi *x-ray* paru-paru digunakan algoritme *K-Means* dan algoritme *Geometric Active Contour*. Pengujian sistem yang digunakan adalah *Receiver Operating Characteristics*, sehingga didapatkan nilai akurasi sebesar 90,03%, sensitifitas

sebesar 62,05%, dan spesifitas sebesar 94,62% untuk paru-paru kiri, sedangkan akurasi sebesar 88,35%, sensitifitas sebesar 63,72%, dan spesifitas sebesar 93,48% untuk paru-paru kanan.

Penelitian terkait dilakukan oleh I Made Oka Widyantar, Agus Tommy Adi Prawira Kusuma, dan Ni Made Ary Esta Dewi Wirastuti (2015), Universitas Udayana, Bali dengan judul “*Preprocessing Pada Segmentasi Citra Paru-Paru dan Jantung Menggunakan Anisotropic Diffusion Filter*”. Pada Penelitian tersebut algoritme *K-Means* dengan nilai masukan  $k = 2$  digunakan pada tahap segmentasinya. Sedangkan pada tahap *preprocessing*-nya, dilakukan normalisasi citra menjadi ukuran  $256 \times 256$  pixel pada tingkat keabuan 256. Pada proses *filter* ini, organ-organ pada citra *x-ray*, seperti pembuluh darah, tulang rusuk, dan tulang selangka dikaburkan sehingga bagian paru-paru terlihat lebih jelas dibandingkan bagian organ yang lain. Optimasi ini disebut dengan metode *Anisotropic Diffusion Filter*. Pengujian yang dilakukan terhadap hasil segmentasi oleh sistem tersebut divalidasi dengan metode *Receiver Operating Characteristics*, didapatkan nilai akurasi sebesar 92,87%, sensitifitas sebesar 89,38%, dan spesifitas sebesar 93,44% untuk paru-paru kiri, sedangkan akurasi sebesar 92,47%, sensitifitas sebesar 85,59%, dan spesifitas sebesar 93,89% untuk paru-paru kanan.

Sebuah penelitian dilakukan oleh Imran Fareed Nizami, Saad UI Hasan, dan Ibrahim Tariq Javed (2014), *Departmen of Electrical Engineering*, Bahria University, Islamabad, Pakistan, yang berjudul “*A Wavelet Frames + K-Means based Automatic Method for Lung Area Segmentation in Multiple Slices of CT Scan*”. Pada penelitian tersebut, metode *K-Means* digunakan untuk proses segmentasinya, dengan menerapkan *background removal* setelah proses segmentasi

tersebut. Proses penghapusan bagian latar belakang pada *CT Scan* dalam bentuk *Wavelet Packet Frames (WPF)* yang dikombinasikan dengan menggunakan algoritme klastering *K-Means* tersebut dapat memberikan hasil segmentasi yang optimal, di mana yang pada umumnya organ trakea masih termasuk dalam hasil segmentasinya menjadi terhapus setelah di segmentasi dengan penerapan *background removal*. Penulis menggunakan *background removal* pada penelitian ini agar dapat meningkatkan nilai akurasi, spesifitas, dan sensitifitas yang lebih optimal apabila dibandingkan dengan penelitian terdahulu.

Penelitian tentang algoritme *EK-Mean* telah dilakukan oleh P.B. Sangamithraa dan S. Govindaraju (2016), *Kumaguru College of Technology*, Coimbatore, India, yang berjudul "*Lung Tumour Detection and Classification using EK-Mean Clustering*". Pada penelitian tersebut, algoritme *K-Means* digunakan untuk klastering citra *CT Images*, yang kemudian hasil klastering tersebut digunakan pada proses algoritme *EK-Mean*, sehingga didapatkan nilai *entropy*, *contrast*, korelasi, homogenitas, dan sebagian dari segmentasi citra pada perhitungan *Fuzzy EK-Mean* yang menunjukkan bagian dari paru-paru yang memiliki tumor.

## **BAB III**

### **PERANCANGAN DAN IMPLEMENTASI SISTEM**

Bab ini menjelaskan mengenai perancangan dari sistem yang meliputi pengumpulan data dan proses pembuatan sistem hingga implementasi sistem. Sistem yang dibangun adalah aplikasi segmentasi citra *x-ray thorax* menggunakan algoritme *K-Means*.

#### **3.1 Studi Literatur**

Literatur yang telah dikaji pada bagian penelitian terkait pada Bab II sebelumnya akan menjadi dasar teori atas studi dan penelitian yang dilakukan. Agar dapat menjadi landasan teori atas dilaksanakannya penelitian ini. Konsep dan dasar teori yang digunakan dapat meliputi sumber buku, jurnal dari penelitian yang telah dilakukan sebelumnya yang relevan. Berikut adalah dasar teori yang digunakan pada penelitian ini:

1. Implementasi sistem perangkat lunak segmentasi citra *x-ray* paru-paru.
2. *Background removal* pada citra *x-ray* paru-paru.
3. Algoritme *K-Means*.
4. Segmentasi citra *x-ray* paru-paru.
5. Metode pengujian akurasi *Receiver Operating Characteristics*.

#### **3.2 Pengumpulan Data**

Data yang dikumpulkan adalah data citra *x-ray* dari *thorax* atau paru-paru pasien yang kemudian akan digunakan untuk implementasi. Data yang dikumpulkan hanya terdapat dua jenis data, yaitu data citra *x-ray thorax* yang



disegmentasi secara manual dan data citra *x-ray thorax* yang disementasi menggunakan sistem. Data-data yang dikumpulkan dapat dilihat pada Tabel 3.1.1

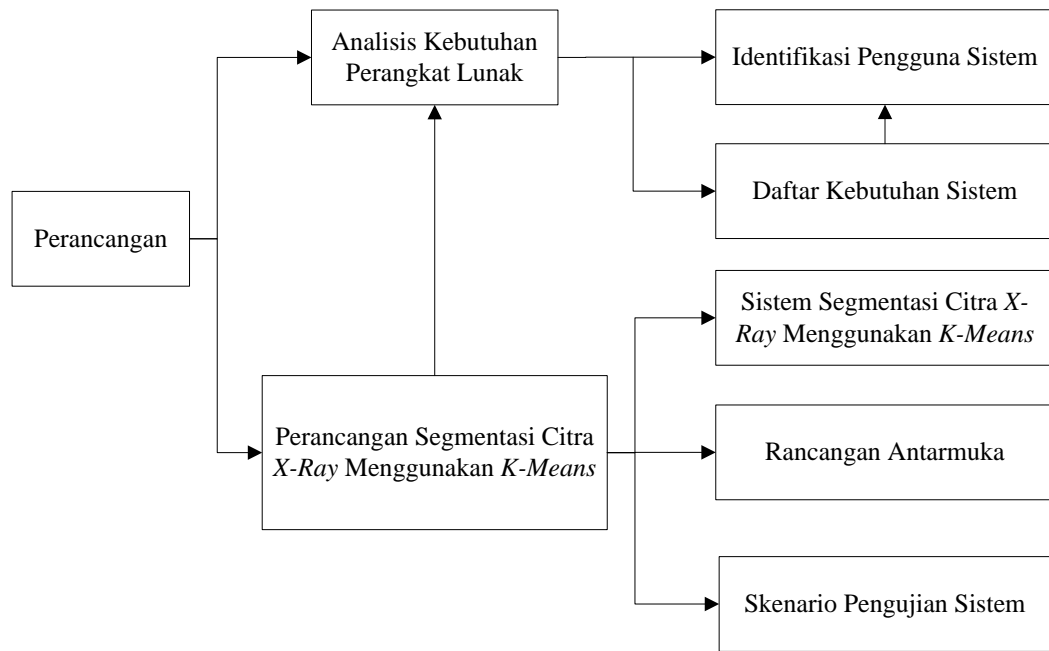
**Tabel 3.1** Kebutuhan Data

No.	Kebutuhan Data	Sumber Data	Metode	Kegunaan Data
1	Data citra <i>x-ray thorax</i>	<a href="http://www.isi.uu.nl/Research/Databases/SCR/">http://www.isi.uu.nl/Research/Databases/SCR/</a>	<i>Download</i>	Data citra <i>x-ray</i> untuk diolah oleh sistem.
2	Segmentasi manual data citra <i>x-ray thorax</i>	Data citra <i>x-ray thorax</i>	<i>Adobe Photoshop (Magic Wand)</i>	Pembanding pada pengujian sistem.

### 3.3 Perancangan Sistem

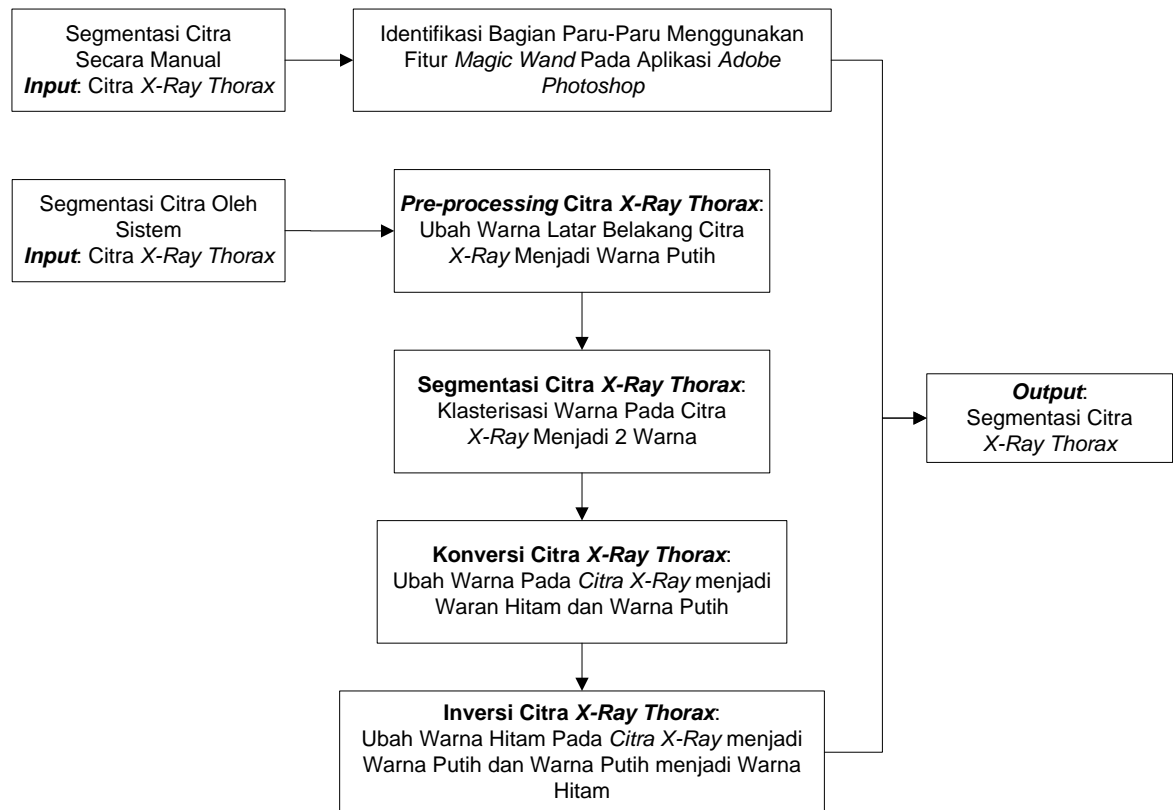
Pada bagian perancangan sistem ini akan dibahas tahap-tahap yang diperlukan sistem untuk melakukan implementasi segmentasi paru-paru pada citra *x-ray thorax* menggunakan *K-Means*. Bagian ini juga membahas kebutuhan aplikasi, sehingga perancangan yang disusun ini dapat diterapkan yang kemudian akan dibahas pada bab berikutnya. Gambar 3.1 menampilkan diagram perancangan secara keseluruhan yang dilakukan pada penelitian ini.

Perancangan untuk alur sistem memiliki tiga tahap bagian yang menjadi pokok utama. Yaitu tahap *pre-processing*, segmentasi menggunakan algoritme *K-Means*, konversi citra hasil segmentasi menjadi biner, dan tahap akhir adalah inversi pada citra segmentasi.



**Gambar 3.1** Diagram Perancangan

Dalam rancangan sistem, terdapat dua proses yang akan dilakukan untuk melakukan segmentasi citra *x-ray* menggunakan *K-Means*, yaitu proses segmentasi citra *x-ray* secara manual dan proses segmentasi citra *x-ray* oleh sistem menggunakan algoritme *K-Means*. Proses segmentasi citra *x-ray* secara manual dilakukan dengan menggunakan aplikasi *Adobe Photoshop*, di mana proses ini dilakukan untuk mendapatkan hasil yang paling ideal pada proses segmentasi citra *x-ray*. Sehingga, pada proses segmentasi manual, citra yang dihasilkan hanya menunjukkan bagian organ *thorax*. Setelah proses segmentasi secara manual dilakukan, maka dilanjutkan dengan proses segmentasi oleh sistem. Proses segmentasi citra *x-ray* menggunakan *K-Means* digunakan untuk mengidentifikasi bagian *thorax* yang ada pada citra *x-ray* dengan menggunakan algoritme *K-Means* untuk melakukan pengelompokan setiap organnya. Blok diagram sistem yang menunjukkan alur dari sistem yang akan dibangun ditampilkan pada Gambar 3.2.



**Gambar 3.2** Diagram Blok Sistem

### 3.4 Analisis Kebutuhan Perangkat Lunak

Sistem yang diimplementasikan dilakukan analisis untuk penentuan kebutuhannya. Analisis yang telah dilakukan didasari oleh rancang bangun aplikasi untuk penerapan segmentasi paru-paru pada citra *x-ray thorax* menggunakan *K-Means*. Analisis kebutuhan perangkat lunak disusun sebagai acuan pada bentuk model pada informasi yang disusun. Analisis kebutuhan perangkat lunak yang dibutuhkan pada penelitian ini adalah identifikasi pengguna sistem, dan daftar kebutuhan sistem.

### 3.4.1 Identifikasi Pengguna Sistem

Identifikasi pengguna dilakukan untuk menetapkan siapa saja yang akan menjadi pengguna aplikasi, agar dapat berinteraksi dengan system. Tabel 3.2 menampilkan penjelasan siapa saja yang memiliki peran dalam penggunaan sistem rekomendasi hunian ideal untuk mencari peringkat rumah yang ideal yang dilengkapi dengan penjelasan deskripsi dari setiap tipe pengguna.

**Tabel 3.2** Identifikasi Pengguna Sistem

No.	Tipe	Deskripsi
1	Dokter paru	Dokter paru merupakan aktor yang melakukan diagnosis penyakit paru berdasarkan <i>x-ray thorax</i> .
2	<i>IT service</i>	<i>IT Service</i> merupakan aktor yang merancang dan membangun sistem, serta melakukan perawatan serta pembaharuan terhadap basis pengetahuan dan data-data sistem secara keseluruhan agar sistem memiliki hasil yang optimal.

### 3.4.2 Daftar Kebutuhan Sistem

Daftar kebutuhan sistem mencakup kebutuhan yang harus terpenuhi oleh sistem supaya pengguna melakukan aplikasi. Pada tabel 3.3 menampilkan daftar kebutuhan yang dimiliki oleh sistem.

**Tabel 3.3** Daftar Kebutuhan Sistem

No.	Kebutuhan	User	Tindakan
1	Tampilan untuk melakukan segmentasi citra <i>x-ray thorax</i> .	Dokter Paru, <i>IT Service</i>	Segmentasi citra <i>x-ray thorax</i> .
2	Tampilan data segmentasi citra <i>x-ray thorax</i> secara manual untuk digunakan pada pengujian sistem.	Dokter Paru, <i>IT Service</i>	Menampilkan citra <i>x-ray thorax</i> yang disegmentasi secara manual.
3	Tampilan pengujian hasil segmentasi citra <i>x-ray thorax</i> oleh sistem	Dokter Paru, <i>IT Service</i>	Menampilkan pengujian hasil Segmentasi citra <i>x-</i>

			<i>ray thorax</i> oleh sistem.
--	--	--	--------------------------------

### 3.5 Sistem Segmentasi Citra *X-Ray* Menggunakan *K-Means*

Berikut adalah langkah-langkah dari rangkaian proses yang akan diimplementasi menjadi sebuah sistem perangkat lunak segmentasi paru-paru pada citra *x-ray thorax* menggunakan *K-Means*.

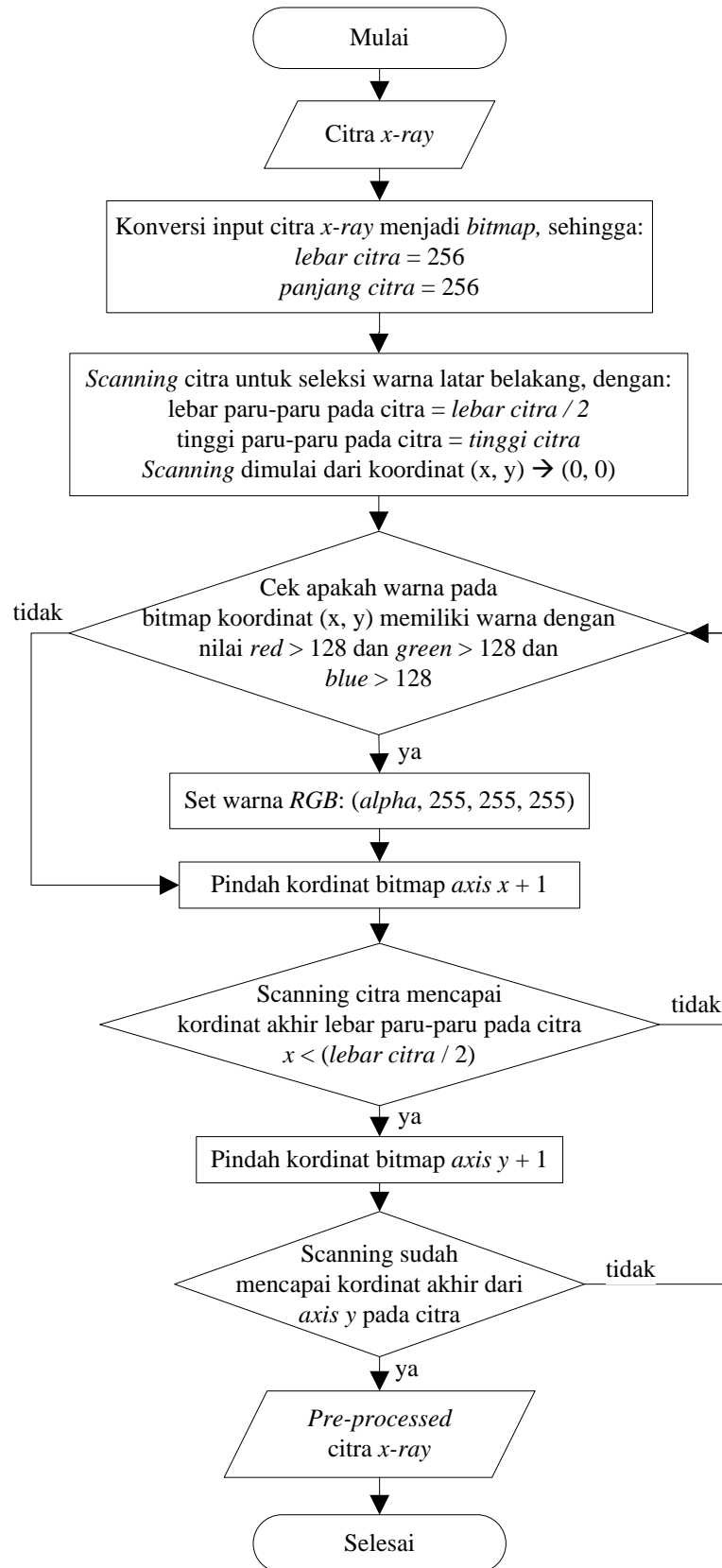
#### Langkah 1: Pre-Processing Citra *X-Ray*

Langkah pertama yang dilakukan oleh sistem adalah mengambil citra *x-ray* dan mengubah formatnya menjadi bitmap dengan ukuran  $256 \times 256$  *pixel*. Kemudian menghapus latar belakang dari citra *x-ray*. Latar belakang pada citra *x-ray* pada umumnya adalah berwarna hitam. Contoh dari citra *x-ray thorax* ditampilkan pada Gambar 3.3.

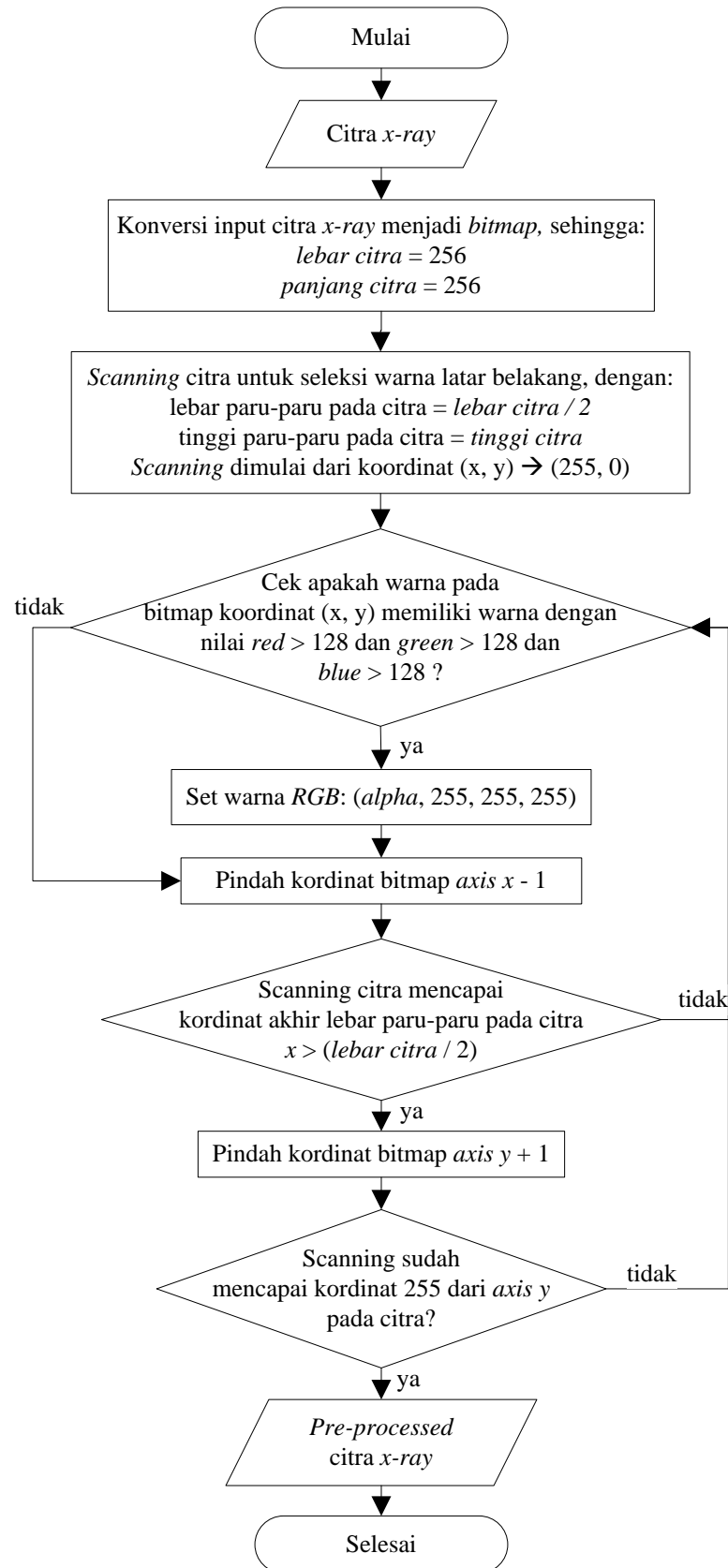


**Gambar 3.3** Citra *X-Ray* Paru-Paru

Setelah didapatkan citra *x-ray* dan ubah *format*-nya menjadi bitmap, selanjutnya adalah dilakukan *pre-processing image*, dengan cara menghapus latar belakang dari citra *x-ray* tersebut. Proses dari penghapusan latar belakang citra *x-ray thorax* ditampilkan pada Gambar 3.4 untuk paru-paru kirin dan Gambar 3.5 untuk paru-paru kanan.

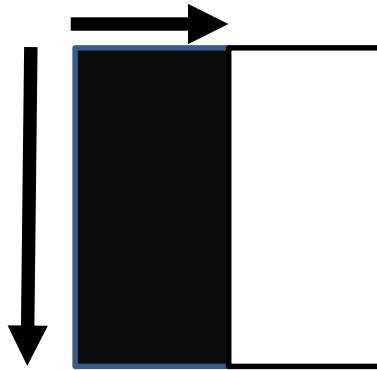


**Gambar 3.4** Diagram Alir *Pre-processing* Citra X-Ray Paru-Paru Kiri

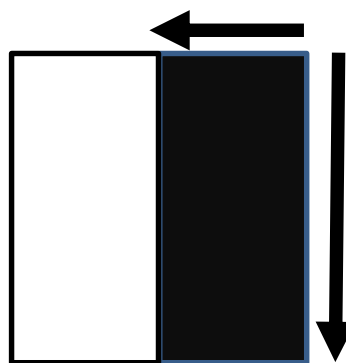


**Gambar 3.5** Diagram Alir *Pre-processing* Citra X-Ray Paru-Paru Kanan

Berdasarkan Gambar 3.4 dan Gambar 3.5, setelah citra *x-ray* ditentukan, citra *x-ray* kemudian akan dikonversi menjadi tipe data *bitmap* dan diambil panjang lebar dan tingginya. Namun karena pada proses ini paru-paru kanan dan paru-paru kiri dilakukan secara terpisah, maka lebar dari citra *bitmap* dibagi dengan 2, sehingga proses *scanning* untuk paru-paru kiri dimulai dari kordinat (0, 0) sampai pada kordinat (128, 255) ditunjukkan pada Gambar 3.6, sedangkan proses *scanning* untuk paru-paru kanan dimulai dari kordinat (255, 0) sampai pada kordinat (128, 255) ditunjukkan pada Gambar 3.7.



**Gambar 3.6** Proses *scanning* warna latar belakang citra *x-ray* paru-paru kiri



**Gambar 3.7** Proses *scanning* warna latar belakang citra *x-ray* paru-paru

kanan



Di dalam proses scanning tersebut, warna dengan nilai *RGB* di atas (128, 128, 128) sebagai indicator warna yang ditetapkan sebagai syarat untuk *scanning* tetap berlanjut, namun apabila di tengah proses berjalan ditemukan warna dengan nilai *RGB* di atas (128, 128, 128), maka pengubahan warna latar belakang dari warna gelap ( $RGB > (128, 128, 128)$ ) menjadi putih dihentikan. Proses ini sama diterapkan baik pada citra untuk area paru-paru kanan maupun area paru-paru kiri.

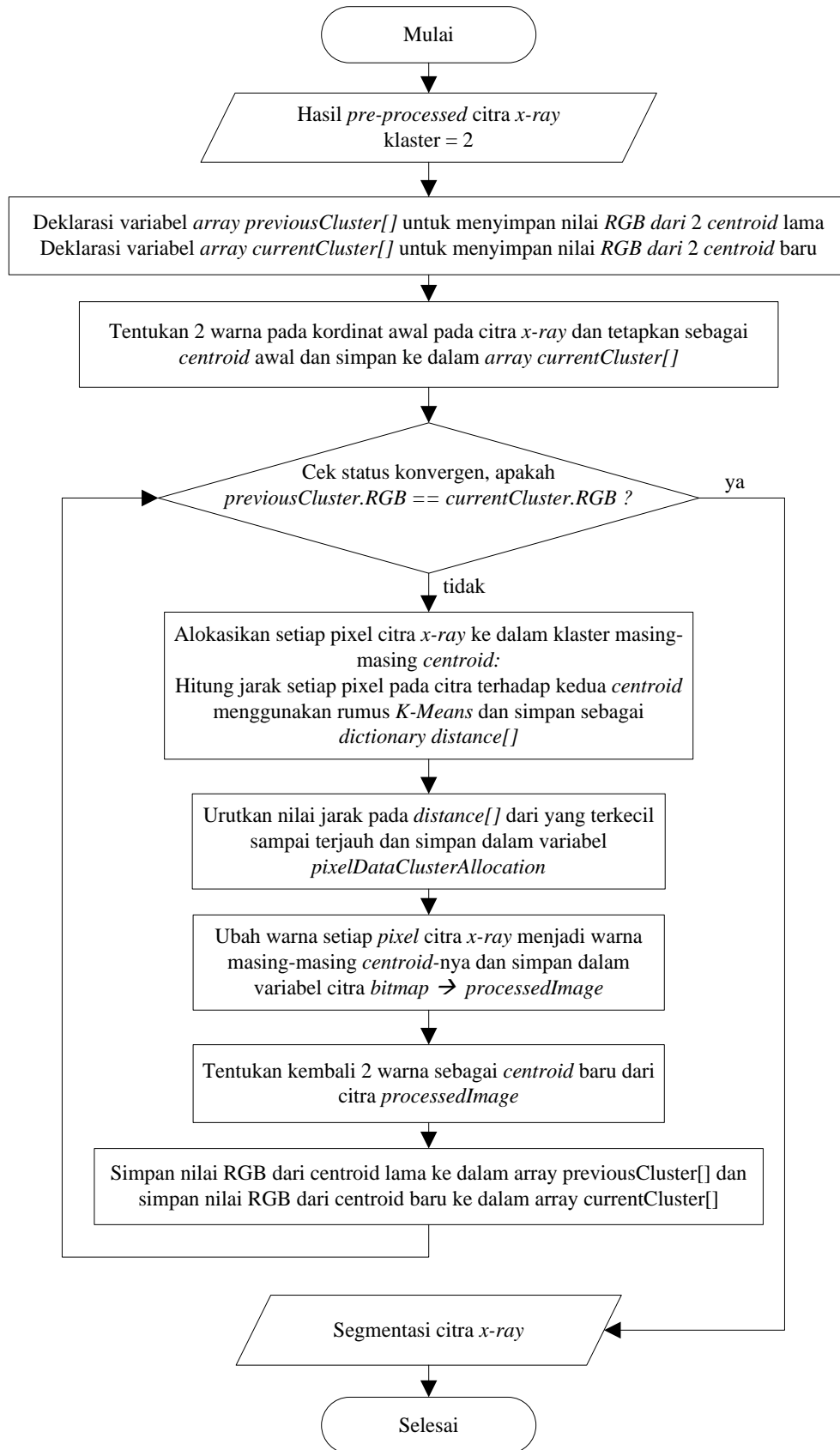
Gambar 3.8 adalah hasil proses penghapusan latar belakang dari citra *x-ray* dari gambar Gambar 3.3.



**Gambar 3.8** Filter Latar Belakang Citra *X-Ray* Paru-Paru

### **Langkah 2: Klastering Citra *X-Ray* Menggunakan Algoritme *K-Means***

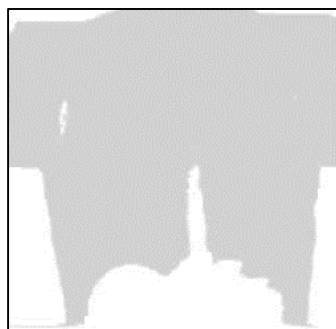
Setelah didapatkan citra *x-ray* yang telah di *pre-processing*, selanjutnya adalah memulai proses segmentasi dengan menggunakan algoritme *K-Means*. Berdasarkan diagram alir pada Gambar 2.5, proses segmentasi citra *x-ray* dengan menggunakan algoritme *K-Means* dibagi menjadi 5 proses utama, sehingga apabila diterapkan pada segmentasi citra *x-ray* thorax yang digunakan pada penelitian ini yang lebih detail ditunjukkan pada Gambar 3.9.



**Gambar 3.9** Diagram Alir Segmentasi Citra X-Ray Thorax

Berdasarkan Gambar 3.9 proses segmentasi citra *x-ray thorax* yang dilakukan pada penelitian ini apabila diterapkan untuk implementasi adalah menjadi sebagai berikut:

1. Memasukkan nilai masukan 2 untuk jumlah kluster warna yang digunakan untuk proses segmentasi. Dalam proses segmentasi ini digunakan hanya 2 kluster warna, dikarenakan untuk didapatkan hasil yang optimal, citra hasil segmentasi diupayakan sebisa mungkin hanya menggambarkan bagian paru-paru saja. Sehingga dengan hanya dua kluster akan mempercepat proses penentuan gambar paru-paru pada hasil segmentasinya.
2. Menentukan pusat kluster (*centroid*). Cara untuk menentukan pusat kluster adalah dengan melakukan pendataan jumlah jenis warna yang ada pada citra, dalam kasus ini adalah Gambar 3.8. Setelah itu serangkaian nilai warna unik yang ada pada citra tersebut di simpan ke dalam variabel *array* dan diurutkan dari warna yang paling tinggi (Semakin mendekati *RGB* [0, 0, 0] berarti semakin tinggi). Gambar 3.10 menampilkan penentuan pusat kluster yang pertama.



**Gambar 3.10** Penentuan Dua Pusat Kluster Pertama Citra *X-Ray*

3. Menghitung jarak setiap data/ objek pada kluster terdekat dengan cara membuat sistem *scanning* secara berulang (iterasi).

- a. Data yang digunakan adalah jumlah dan jenis alokasi warna setiap klaster pada yang ada saat ini. Kemudian cara menghitung jarak setiap *pixel* dengan pusat klaster adalah dengan menggunakan *Math.SQRT* sehingga didapatkan selisih nilai *RGB*-nya sesuai dengan *Euclidean Distance* pada Persamaan 2.1.
  - b. Setelah didapatkan nilai jarak dari masing-masing *pixel* pada citra dengan masing-masing pusat klasternya. Alokasikan setiap warna *pixel* pada citra dengan pusat klaster warna terdekatnya. Sehingga warna dari *pixel* tersebut diubah menjadi warna yang sama dengan pusat klasternya.
4. Menghitung kembali nilai rata-rata warna untuk pusat klaster (*centroid*) yang baru. Caranya adalah dengan mengambil jumlah warna unik yang ada pada pusat klaster sebelumnya, kemudian dicari rata-rata jaraknya kembali untuk masing-masing nilai *RGB*-nya.
  5. Lakukan cek kondisi konvergen.
    - a. Kondisi konvergen adalah di mana setelah menentukan pusat klaster baru, jumlah warna unik untuk setiap pusat klaster yang ada sama dengan pusat-pusat klaster sebelumnya, apabila kondisi ini terpenuhi, maka serangkaian proses klastering 2 warna pada citra *x-ray* dihentikan. Sedangkan apabila kondisi konvergen ini tidak terpenuhi, maka akan dilakukan proses pengulangan penentuan pusat klaster kembali.

- b. Hasil dari proses ini disebut dengan segmentasi citra *x-ray* dan kemudian di simpan hasil segmentasi citra *x-ray* tersebut pada sistem.

Gambar 3.10 merupakan iterasi pertama dari proses segmentasi ini, namun karena pada Gambar 3. 10 masih ditemukan warna-warna unik yang lain, maka proses segmentasi masih dilanjutkan pada Gambar 3.11 untuk iterasi kedua, Gambar 3.12 untuk iterasi ketiga, Gambar 3.13 untuk iterasi keempat, Gambar 3.14 untuk iterasi kelima, Gambar 3.15 untuk iterasi keenam, Gambar 3.16 untuk iterasi ketujuh. Sedangkan Gambar 3.17 adalah iterasi kedelapan di mana kondisi konvergensi telah terpenuhi. Sehingga pusat klaster pada Gambar 3.17 tidak mengalami perubahan titik warna setelah dilakukan perhitungan alokasi pusat klaster baru. Oleh karena itu pada Gambar 3.16 tidak memiliki perbedaan dengan Gambar 3.17.



**Gambar 3.11** Alokasi Pusat Klaster Iterasi Ke-2



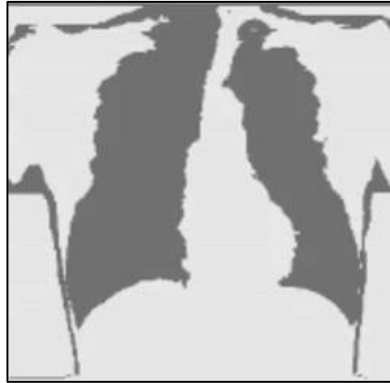
**Gambar 3.12** Alokasi Pusat Klaster Iterasi Ke-3



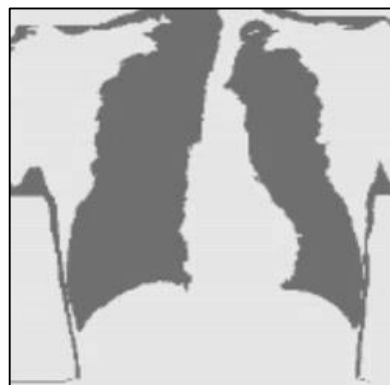
**Gambar 3.13** Alokasi Pusat Klaster Iterasi Ke-4



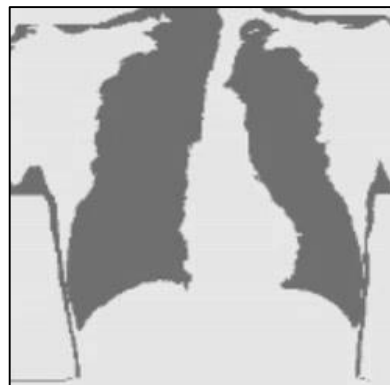
**Gambar 3.14** Alokasi Pusat Klaster Iterasi Ke-5



**Gambar 3.15** Alokasi Pusat Kluster Iterasi Ke-6



**Gambar 3.16** Alokasi Pusat Kluster Iterasi Ke-7



**Gambar 3.17** Alokasi Pusat Kluster Iterasi Ke-8 (Kondisi Konvergen)

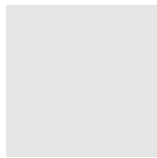

Gambar 3.17 adalah hasil dari penentuan pusat kluster iterasi ke-8 di mana pada iterasi ini kondisi konvergen telah terpenuhi, sehingga proses segmentasi citra *x-ray* dihentikan. Sehingga Gambar 3.17 adalah hasil akhir dari proses segmentasi citra *x-ray* dengan menggunakan algoritme *K-Means*.

### Langkah 3: Konversi Hasil Segmentasi Citra X-Ray Menjadi Biner

Konversi citra diterapkan pada hasil segmentasi citra *x-ray* dengan cara menyaring dua warna yang ada, kemudian mengubah warna yang paling gelap menjadi warna hitam (*RGB*: 0, 0, 0) dan warna yang paling terang menjadi warna putih (*RGB*: 255, 255, 255). Konversi warna ini diperlukan agar mempermudah penulis dalam melakukan perbandingan dengan citra *x-ray* yang disegmentasi secara manual, di mana citra *x-ray* yang disegmentasi secara manual tersebut hanya memiliki dua jenis warna, yaitu hitam (*RGB*: 0, 0, 0) dan putih (*RGB*: 255, 255, 255). Sesuai dengan diagram alir 2.6, proses *scanning* warna dilakukan secara searah dari koordinat *bitmap* (0, 0) hingga koordinat *bitmap* (255, 255).

Citra masukan yang digunakan dalam proses ini adalah Gambar 3.17 yang merupakan hasil segmentasi citra *x-ray* menggunakan algoritme *K-Means*. Pada Gambar 3.17 hanya memiliki 2 kluster warna saja, di mana 2 warna tersebut adalah:

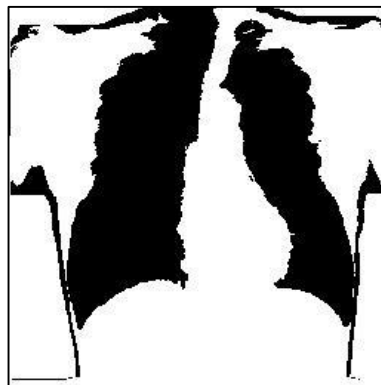
**Tabel 3.4** Daftar Warna Pada Gambar 3.17

No.	Warna	<i>RGB</i>	<i>Hex</i>
1		229, 229, 229	#E5E5E5
2		109, 109, 109	#6D6D6D

Berdasarkan Tabel 3.4, dua jenis warna yang ada pada hasil segmentasi Gambar 3.17 adalah warna putih keabuan dengan *RGB* (229, 229, 229) dan abu-abu kehitaman dengan *RGB* (109, 109, 109). Kemudian kedua warna tersebut



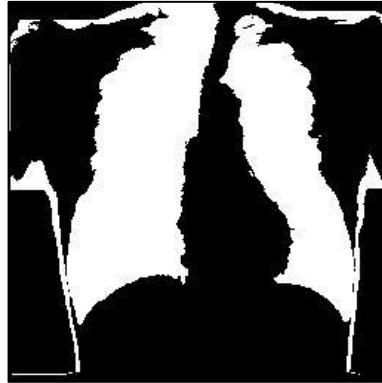
diubah warnanya menjadi hitam dan putih, dengan cara mengganti warna abu kehitaman ( $RGB: 109, 109, 109$ ) menjadi warna hitam ( $RGB: 0, 0, 0$ ) dan warna putih keabuan ( $RGB: 229, 229, 229$ ) menjadi warna putih ( $RGB: 255, 255, 255$ ). Proses ini berjalan secara dinamis, apabila pada citra *x-ray* selanjutnya memiliki warna yang berbeda pada hasil segmentasinya, maka tetap warna yang paling gelap pada hasil segmentasinya dijadikan warna hitam, dan warna yang paling terang dijadikan warna putih. Gambar 3.18 menampilkan hasil dari proses konversi Gambar 3.16 menjadi biner.



**Gambar 3.18** Konversi Hasil Segmentasi Citra X-Ray Menjadi Biner

#### **Langkah 4: Inversi segmentasi citra *x-ray***

Langkah terakhir dari proses segmentasi citra *x-ray* ini adalah dengan melakukan inversi pada hasil segmentasi citra *x-ray* dengan cara menukar warna putih ( $RGB: 255, 255, 255$ ) pada citra menjadi hitam ( $RGB: 0, 0, 0$ ), dan warna hitam ( $RGB: 0, 0, 0$ ) pada citra menjadi warna putih ( $RGB: 255, 255, 255$ ). Proses ini dapat dilakukan karena hasil dari konversi ke biner pada Gambar 3.17 sudah hanya memiliki dua jenis warna saja, yaitu warna putih ( $RGB: 255, 255, 255$ ) dan warna hitam ( $RGB: 0, 0, 0$ ). Hasil dari proses inversi segmentasi citra *x-ray* ditampilkan pada Gambar 3.19.

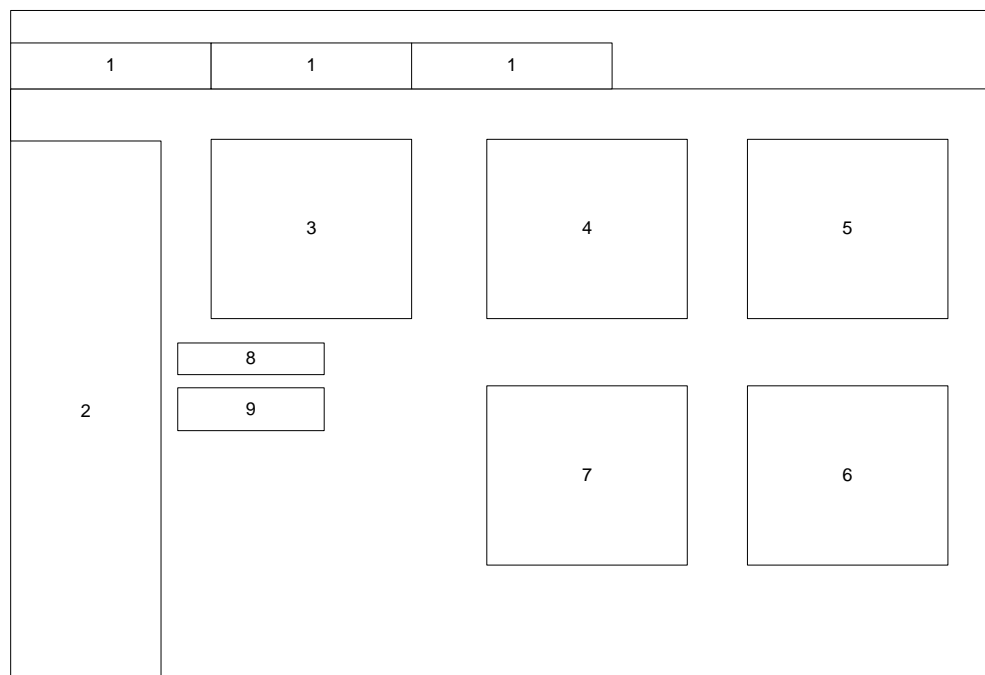


**Gambar 3.19** Inversi Segmentasi Citra *X-Ray*

### 3.6 Rancangan Antarmuka

Rancangan desain antarmuka sistem segmentasi citra *x-ray* akan diterapkan pada platform windows desktop dengan menggunakan bahasa pemrograman C#. Penulis membuat 3 tabulasi pada aplikasi, yaitu tab segmentasi data citra *x-ray*, tab data segmentasi manual, dan tab pengujian. Berikut adalah desain antarmuka yang akan dibangun.

#### 3.6.1 Tab Segmentasi Data *X-Ray*



**Gambar 3.20** Rancangan Tampilan Tab Segmentasi Data *X-Ray*

Halaman segmentasi data *x-ray* yang ditampilkan pada Gambar 3.20 memiliki fitur sebagai berikut:

1. Tabulasi menu yang terdiri dari tab segmentasi citra *x-ray*, tab data segmentasi manual, dan tab pengujian *Receiver Operating Characteristics*.
2. Tabel yang menampilkan data-data asli citra *x-ray* yang telah disimpan pada sistem.
3. Kotak gambar yang menampilkan gambar asli citra *x-ray* sebelum dilakukan proses segmentasi.
4. Kotak gambar yang menampilkan hasil dari proses penghapusan latar belakang warna hitam dari gambar asli citra *x-ray* sebelum dilakukan proses segmentasi.
5. Kotak gambar yang menampilkan hasil dari proses segmentasi citra *x-ray* dengan menggunakan algoritme *K-Means*.
6. Kotak gambar yang menampilkan hasil dari proses konversi menjadi biner pada citra hasil segmentasi menggunakan algoritme *K-Means*.
7. Kotak gambar yang menampilkan hasil dari proses inversi citra *x-ray* pada citra hasil konversi menjadi biner.
8. Sebuah tombol untuk melakukan proses segmentasi citra *x-ray* menggunakan algoritme *K-Means*.
9. Nilai masukan berupa jumlah klaster yang akan digunakan pada proses segmentasi algoritme *K-Means*. Nilai masukannya adalah permanen, yaitu 2.

### 3.6.1 Tab Data Segmentasi Manual

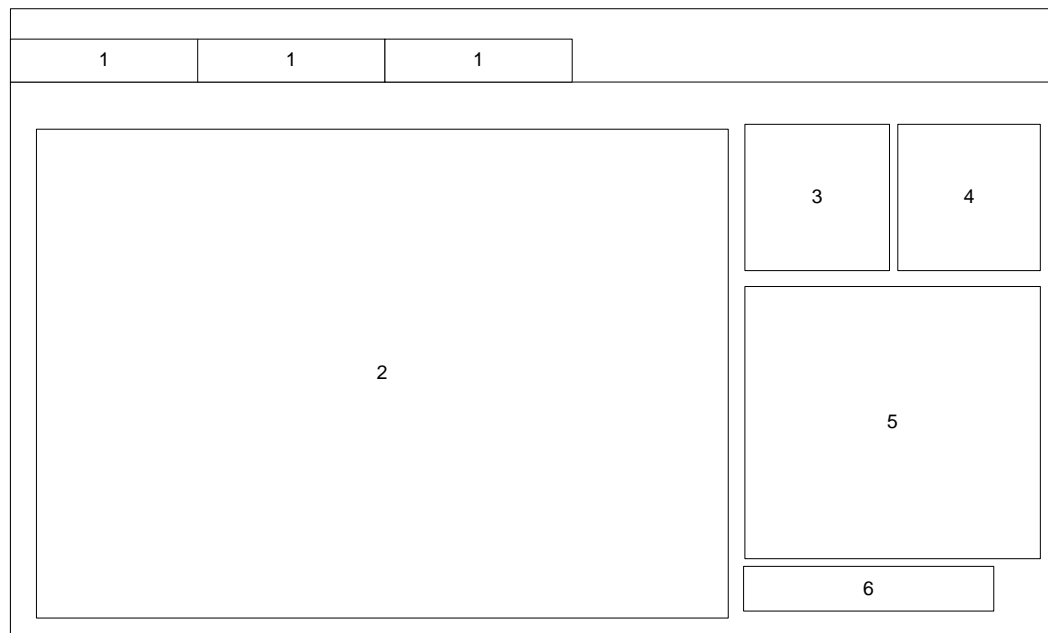
1	1	1		
2	2	2	2	2
2	2	2	2	2
2	2	2	2	2

**Gambar 3.21** Rancangan Tampilan Tab Data Segmentasi Manual

Halaman tampilan data segmentasi manual yang ditampilkan pada Gambar 3.21 memiliki fitur sebagai berikut:

1. Tabulasi menu yang terdiri dari tab segmentasi citra *x-ray*, tab data segmentasi manual, dan tab pengujian *Receiver Operating Characteristics*.
2. Kotak gambar yang menampilkan citra *x-ray* yang telah disegmentasi secara manual.

### 3.6.2 Tab Pengujian



**Gambar 3.22** Rancangan Tampilan Tab Pengujian

Halaman pengujian *Receiver Operating Characteristics* yang ditampilkan pada Gambar 3.22 memiliki fitur sebagai berikut:

1. Tabulasi menu yang terdiri dari tab segmentasi citra *x-ray*, tab data segmentasi manual, dan tab pengujian *Receiver Operating Characteristics*.
2. Tabel yang menampilkan data hasil segmentasi citra *x-ray* oleh sistem.
3. Kotak gambar yang menampilkan gambar hasil segmentasi citra *x-ray* oleh sistem yang kemudian akan disandingkan dengan gambar hasil segmentasi citra *x-ray* secara manual.
4. Kotak gambar yang menampilkan gambar hasil segmentasi citra *x-ray* secara manual yang kemudian akan disandingkan dengan hasil segmentasi citra *x-ray* oleh sistem.

5. Area data list yang menampilkan parameter-parameter yang digunakan pada pengujian *Receiver Operating Characteristics*, seperti nilai *True Positif*, *True Negatif*, *False Positif*, *False Negatif*, akurasi, sensitifitas, dan spesifitas dari dua citra yang ditampilkan pada poin nomor 3 dan poin nomor 4.
6. Sebuah tombol yang digunakan untuk memulai proses perhitungan pengujian dengan menggunakan metode *Receiver Operating Characteristics*.

### **3.7 Rancangan Pengujian Sistem**

Pengujian yang dilakukan terhadap sistem adalah perbandingan hasil segmentasi citra *x-ray* dari sistem dengan hasil segmentasi citra *x-ray* secara manual dengan menggunakan metode *Receiver Operating Characteristics*. Uji sensitivitas, spesifisitas, dan akurasi dilakukan untuk mengetahui kesesuaian dan nilai evaluasi atas penerapan implementasi algoritme *K-Means* yang digunakan pada penelitian ini. Terdapat 4 parameter yang digunakan dalam perhitungannya. Yaitu *True Positif*, *True Negatif*, *False Positif*, *False Negatif*. Dengan menggunakan 4 parameter tersebut, akan diketahui nilai akurasi, spesifitas, dan sensitifitas dari kedua citra *x-ray* yang dibandingkan.

Menurut Baratloo A. et al. (2015), pengujian akurasi citra segmentasi dapat dilihat pada hasil kemampuan dari sistem untuk membedakan bagian pada citra segmentasi sistem yang sesuai dengan citra segmentasi manual. Dengan menghitung positif sejati dan negatif sejati dari pada permukaan citra segmentasi sistem dengan citra segmentasi manual. Pengujian sensitifitas citra merupakan sebuah penilaian sebaik apa sistem dapat menentukan bagian paru-paru yang tidak

sehat pada citra segmentasi sistem dengan citra segmentasi manual. Perhitungan dilakukan dengan cara menghitung jumlah proposi *pixel* yang positif sejati. Pengujian spesifitas citra merupakan sebuah penilaian sebaik apa sistem dapat menentukan bagian paru-paru yang sehat pada citra segmentasi sistem dengan citra segmentasi manual. Perhitungan dilakukan dengan cara menghitung proporsi negatif sejati pada paru-paru yang sehat.

Nilai akurasi, sensitifitas, dan spesifitas dapat dihitung dengan menggunakan Persamaan 3.1 untuk perhitungan akurasi, Persamaan 3.2 untuk perhitungan sensitifitas, dan Persamaan 3.3 untuk perhitungan spesifitas.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(3.1)$$

$$Sensitifitas = \frac{TP}{TP+FN} \dots\dots\dots(3.2)$$

$$Spesifitas = \frac{TN}{TN+FP} \dots\dots\dots(3.3)$$

Keterangan:

TP = *True Positif*, nilai kebenaran antara hasil gambar segmentasi sistem pada paru-paru (dalam jumlah *pixel*).

TN = *True Negatif*, nilai kebenaran antara hasil gambar segmentasi sistem pada latar belakang (dalam jumlah *pixel*).

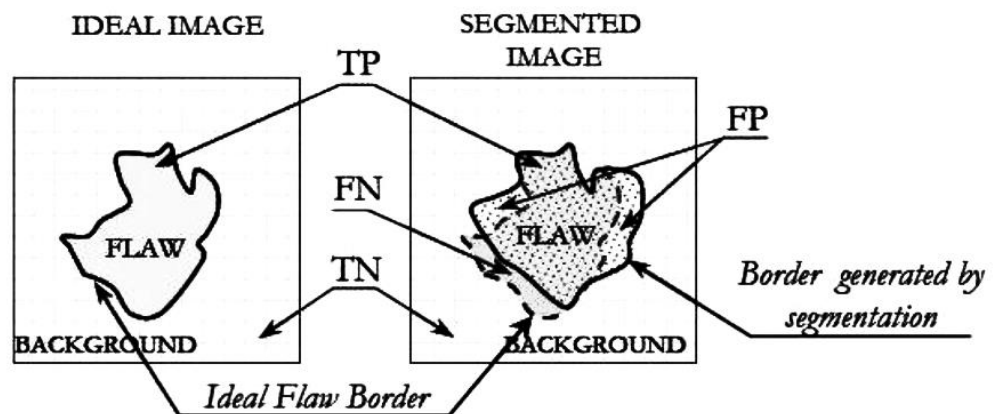
FP = *False Positif*, nilai ketidaktepatan antara hasil gambar segmentasi sistem pada paru-paru (dalam jumlah *pixel*).

FN = *False Negatif*, nilai ketidaktepatan antara hasil gambar segmentasi sistem pada latar belakang (dalam jumlah *pixel*).

Parameter TP, TN, FP, dan FN menghitung jumlah *pixel* yang ada pada citra yang masuk dalam kriteria masing-masing parameter. Keempat parameter ini dihitung sebanyak 45 data citra *x-ray* yang disegmentasi oleh sistem yang dibandingkan dengan citra *x-ray* yang disegmentasi secara manual. Kemudian

dihitung nilai akurasi, sensitifitas, dan spesifitasnya dari setiap perbandingan 45 segmentasi citra *x-ray* tersebut. Kemudian di akhir dihitung rata-rata dari parameter TP, parameter TN, parameter FP, parameter FN nilai akurasi, nilai sensitifitas, dan nilai spesifitasnya dari 45 uji coba segmentasi tersebut.

Gambar 3.23 menampilkan perbedaan antara citra asli dengan citra hasil segmentasi yang ditunjukkan parameter yang digunakan untuk mencari akurasi, sensitifitas, dan spesifitasnya.



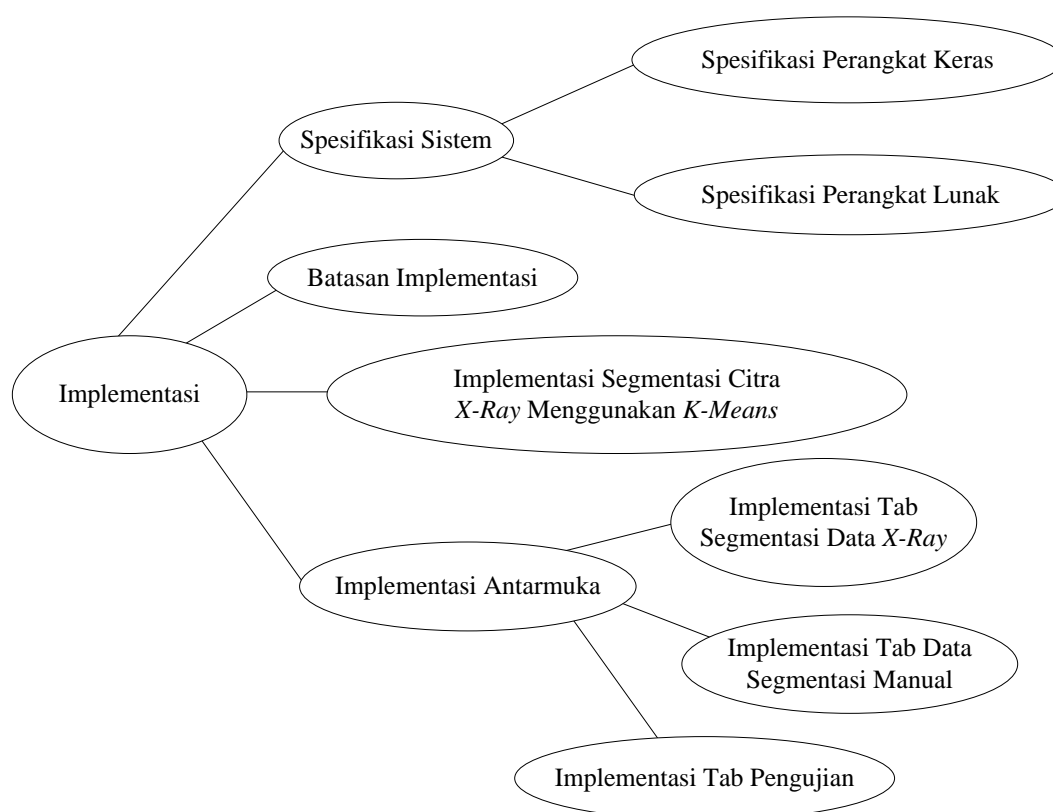
**Gambar 3.23** Perbandingan citra segmentasi manual dengan citra segmentasi sistem



## BAB IV

### IMPLEMENTASI DAN PENGUJIAN

Pada bab ini dibahas mengenai implementasi perangkat lunak berdasarkan perancangan yang telah disusun dengan analisis kebutuhan perangkat lunak dan hasil pengujian yang dilakukan terhadap metode yang digunakan pada penelitian ini. Pada gambar 4.1 ditampilkan konsep dari pengimplementasian yang dilakukan dengan menguraikan spesifikasi perangkat lunak, batasan dari implemetasi, implementasi algoritma *K-Means* pada segmentasi citra *x-ray*, dan implementasi antarmuka perangkat lunak.



**Gambar 4.1** Pohon Implementasi

## 4.1 Spesifikasi Sistem

Spesifikasi sistem meliputi spesifikasi perangkat lunak dan spesifikasi perangkat keras yang digunakan untuk implementasi dan pengembangan perangkat lunak. Spesifikasi sistem ditentukan agar pengembangan dari perangkat lunak sesuai dengan perrancangan yang telah disusun sebelumnya.

### 4.1.1 Spesifikasi Perangkat Keras

Perangkat keras yang digunakan untuk implementasi sistem ditunjukkan seperti pada Tabel 4.1.

**Tabel 4.1** Spesifikasi Perangkat Keras Sistem

No.	Nama Komponen	Spesifikasi
1	<i>System Model</i>	<i>LENOVO IdeaPad 110 Black</i>
2	<i>Chipset</i>	<i>Intel Celeron N3160</i>
3	<i>Memory</i>	<i>2GB DDR3L</i>
4	<i>Harddisk</i>	<i>512 GB HDD</i>

### 4.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan untuk implementasi sistem ditunjukkan seperti pada Tabel 4.2 memiliki spesifikasi yang ditunjukkan seperti pada Tabel 4.2 di bawah ini.

**Tabel 4.2** Spesifikasi Perangkat Lunak Sistem

No.	Komponen	Detail
1	<i>Operating System</i>	<i>Windows 8 32-bit</i>
2	Pemrograman	<i>C# Language Programming</i>
3	<i>Tools</i>	<i>Microsoft Visual Studio 2012</i>

## 4.2 Batasan Implementasi

Batasan implementasi menjelaskan garis batasan untuk fitur apa saja yang dapat dimiliki oleh aplikasi. Batasan diperlukan ini supaya penelitian ini dalam implementasinya memiliki ruang lingkup yang jelas. Sebagai berikut ini adalah batasan implementasi untuk aplikasi:

1. Data masukan yang dapat diterima aplikasi adalah citra *x-ray* dengan sudah jenis warna *grayscale* dengan ukuran resolusi  $256 \times 256$  *pixel*.
2. Data keluaran aplikasi ini adalah citra *x-ray* hasil dari proses segmentasi dengan menggunakan algoritma *K-Means*.
3. Pengguna tidak dapat mengubah nilai *inputan* untuk jumlah kluster (*k*) selain 2.
4. Data yang disimpan dalam sistem hanya sebanyak 45 data citra *x-ray* asli dan 45 data citra segmentasi manual untuk perbandingan pada pengujiannya.
5. Data 45 citra segmentasi manual adalah metode segmentasi dengan menggunakan fitur *Magic Wand* pada aplikasi *Adobe Photoshop* pada setiap data 45 data citra *x-ray* asli.

## 4.3 Implementasi Segmentasi Citra X-Ray Menggunakan Algoritme *K-Means*

Sistem yang dibangun pada penelitian ini menggunakan bahasa pemrograman *C#* berbasis *.Net* pada sistem operasi Windows. Sistem segmentasi citra *x-ray* yang diimplementasikan sesuai pada diagram alir Gambar 2.3 memiliki 4 bagian, yaitu *pre-processing* citra *x-ray*, segmentasi citra menggunakan algoritma *K-Means*, konversi citra hasil segmentasi menjadi biner, dan yang terakhir adalah inversi citra hasil segmentasi.

### 4.3.1 Pre-processing Citra X-Ray

Implementasi *pre-processing* citra *x-ray* diawali dengan proses pengambilan citra *x-ray* oleh sistem dan ditampilkan pada aplikasi. Kemudian dilakukan penghapusan latar belakang (warna hitam *RGB*: 0, 0, 0). Sehingga citra yang dihasilkan dari proses ini adalah tampilan citra *x-ray* paru-paru dengan latar belakang warna putih. Implementasi dari proses ini dalam bentuk *source code* ditampilkan pada Tabel 4.3.

**Tabel 4.3** *Souce Code* Pre-processing Citra X-Ray Paru-Paru

No.	Kode
1	private void Segmentasi_Filter_Background()
2	{
3	Bitmap bmp = new Bitmap(this.pictureBox1.Image);
4	int width = bmp.Width / 2;
5	int height = bmp.Height;
6	int indikator = 0;
7	//SCAN PARU-PARU KIRI
8	for (int y = 0; y < height; y++)
9	{
10	indikator = 0;
11	for (int x = 0; x <= width; x++)
12	{
13	Color colour = bmp.GetPixel(x, y);
14	//ekstraksi nilai ARGB dari variabel colour
15	int alpha = colour.A;
16	int red = colour.R;
17	int green = colour.G;
18	int blue = colour.B;
19	if (red > 128 && green > 128 && blue > 128)

20	{
21	indikator = 1;
22	}
23	if (indikator == 0)
24	{
25	bmp.SetPixel(x, y, Color.FromArgb(alpha, 255, 255, 255));
26	}
27	}
28	}
29	//SCAN PARU-PARU KANAN
30	for (int y = 0; y < height; y++)
31	{
32	indikator = 0;
33	for (int x = 255; x >= width; x--)
34	{
35	Color colour = bmp.GetPixel(x, y);
36	//ekstraksi nilai ARGB dari variabel colour
37	int alpha = colour.A;
38	int red = colour.R;
39	int green = colour.G;
40	int blue = colour.B;
41	if (red > 128 && green > 128 && blue > 128)
42	{
43	indikator = 1;
44	}
45	
46	if (indikator == 0)
47	{

48	<code>bmp.SetPixel(x, y, Color.FromArgb(alpha, 255, 255, 255));</code>
49	<code>}</code>
50	<code>}</code>
51	<code>}</code>
52	<code>this.pictureBox8.Image = bmp;</code>
53	<code>this.Save_Img_Filter_Background();</code>
54	<code>}</code>

Tabel 4.3 menampilkan sebuah *method source code* yang melakukan pengambilan data citra *x-ray* kemudian melakukan *scanning* latar belakang untuk warna hitam saja. Tabel 4.4 menerangkan penjelasan dari masing-masing perintah setiap barisnya.

**Tabel 4.4** Penjelasan Fungsi Dari Setiap Perintah Tabel 4.3

Baris	Fungsi
1	Deklarasi <i>method</i> <code>Segmentasi_Filter_Background()</code>
3	Mengambil citra yang telah di pilih dan ditampilkan pada <i>picturebox1</i> sebagai citra <i>inputan</i> untuk proses ini. Dan dikonversi menjadi <i>bitmap</i> .
4	Menentukan lebar <i>bitmap</i> dibagi dengan dua, untuk membedakan paru-paru kanan dan paru-paru kiri.
5	Menentukan tinggi <i>bitmap</i> .
6	Indikator untuk syarat dilakukan perubahan warna, apabila nilainya 0 berarti warna pada kordinat tersebut adalah hitam, maka diubah menjadi putih. Apabila nilainya 1 berarti warna pada kordinat tersebut adalah putih, maka tidak ada perubahan warna.
8	Inisialisasi looping kordinat <i>bitmap</i> dimulai dari <i>axis y</i> pada titik ke 0 sampai ke titik 255.
11	Inisialisasi looping kordinat <i>bitmap</i> dimulai dari <i>axis x</i> pada titik ke 0 sampai ke titik 127.
13-18	Pengambilan warna pada kordinat saat ini, yang diambil adalah nilai aRGB.

19-22	Inisialisasi kondisi, apabila nilai RGB kordinat saat ini lebih dari 128, maka indikator di-set nilainya menjadi 1.
23-26	Inisialisasi kondisi, apabila nilai indikator masih 0, maka pada iterasi kordinat saat ini diubah warnanya menjadi putih ( <i>RGB: 255, 255, 255</i> ).
29	Memulai proses <i>scanning</i> latar belakang untuk paru-paru kanan.
30	Inisialisasi looping kordinat <i>bitmap</i> dimulai dari <i>axis y</i> pada titik ke 0 sampai ke titik 255.
33	Inisialisasi looping kordinat <i>bitmap</i> dimulai dari <i>axis x</i> pada titik ke 255 sampai ke titik 128.
35-40	Pengambilan warna pada kordinat saat ini, yang diambil adalah nilai aRGB.
41-43	Inisialisasi kondisi, apabila nilai RGB kordinat saat ini lebih dari 128, maka indikator di-set nilainya menjadi 1.
46-49	Inisialisasi kondisi, apabila nilai indikator masih 0, maka pada iterasi kordinat saat ini diubah warnanya menjadi putih ( <i>RGB: 255, 255, 255</i> ).
52	Tampilkan hasil citra dari proses ini pada <i>pictureBox8</i> .
53	Simpan hasil citra dari proses ini di dalam sistem.

#### 4.3.2 Segmentasi Citra X-Ray Menggunakan Algoritme K-Means

Berdasarkan Gambar 3.9, segmentasi citra *x-ray* menggunakan algoritme *K-Means* yang kemudian dijabarkan menjadi 5 langkah proses dari menentukan centroid warna berdasarkan nilai masukan dua kluster hingga proses cek status konvergen. Berikut adalah penjelasan langkah implementasi segmentasi lebih detail berdasarkan konsep segmentasi pada Gambar 3.9:

1. Proses segmentasi pada *source code* di mulai dari inisialisasi *object* kelas *KMeans* baru dengan 3 parameter *input*, yaitu citra *x-ray* hasil *pre-processing*, 2 kluster, dan model citra RGB. Ketiga parameter *input* tersebut menjadi sebuah proses pemanggilan dari *method* untuk memulai

segmentasi citra tahap awal untuk mencari dua warna *centroid* awal dalam masing-masing dalam nilai RGB.

2. Untuk memulai proses mencari dua *centroid* awal, perlu dilakukan penyimpanan citra ke dalam sistem memori dengan menggunakan inisialisasi metode *unsafebitmap* dengan kode *lockbitmap* untuk alokasi alamat setiap *pixel* dari citra pada sistem memori untuk dilakukan *scanning* RGB dan kode *unlockbitmap* untuk melepaskan (menghapus) alokasi alamat setiap *pixel* dari citra. Proses penyimpanan data *pixel* dari citra ini dilakukan untuk setiap akan dilakukan proses *scanning* citra untuk diambil nilai RGB nya. Berbeda dengan proses *pre-processing*, proses segmentasi menggunakan data *pixel* dari citra karena penyimpanan data *pixel* dilakukan pada sistem memori.
3. Proses untuk mencari warna dua *centroid* awal dilakukan dengan melakukan *scanning* jenis warna apa saja yang ada pada citra kemudian disebut dengan *method FindTopXColours(numCluster)*. Parameter input *numCluster* adalah jumlah klaster yang akan ditetapkan untuk klasterisasi warna pada citra, yang dalam hal ini nilainya adalah 2 klaster. *Method* ini kemudian mencatat nama warna unik apa saja yang ada pada citra *x-ray*. Nama warna yang digunakan adalah nama warna yang telah didefinisikan secara *default* pada *System.drawing* untuk .NET Framework Windows sesuai ketentuan pemrograman bahasa C#. Setelah didapatkan semua nama uniknya dari citra, kemudian akan disimpan dalam variabel *dictionary colours* dengan instance object dari kelas *ColourCount*. Kelas *ColourCount* tersebut menyimpan atribut nama warna (variabel *\_clr*) dan



urutan warna (variabel *\_count*). Penggunaan tanda “\_” (*underscore*) pada sebelum penamaan variabel berarti variabel tersebut bersifat *private* yang hanya bisa diakses dari kelas yang menaungi variabel tersebut masing-masing.

4. Deklarasi variabel *array \_topColours* yang berfungsi untuk menyimpan dua warna klaster yang berisikan nilai RGB masing-masing. Variabel ini yang menyimpan nilai yang kemudian disebut dengan dua *centroid* warna citra. Nilai RGB yang tersimpan di dalamnya didapatkan dengan cara melakukan *sorting* dari id dari nama warna yang telah disimpan pada variabel *dictionary colours*.
5. Kemudian simpan setiap nilai RGB dari dua *centroid* awal citra ke dalam variabel *dictionary \_previousCluster* dan variabel *dictionary \_currentCluster*. Kedua variabel *dictionary* ini kemudian akan digunakan untuk proses iterasi alokasi warna setiap *pixel* ke dalam *centroid* warna terdekat masing-masing.
6. Tahap selanjutnya adalah melakukan iterasi yang berisikan komponen *method Iterate()* untuk melakukan proses alokasi warna setiap *pixel* ke dalam klasternya masing-masing dan menentukan kembali *centroid* baru. Tahap iterasi ini akan diulang secara terus menerus dalam konsep fungsi *timer* yang telah disediakan dalam .NET Framework pemrograman C#. Fungsi *timer* ini digunakan karena proses segmentasi menggunakan penyimpanan data *pixel* ke dalam sistem memori (RAM).
7. Di dalam *method timer\_tick*, terdapat *method Iterate()* dan fungsi pemeriksaan status konvergen menggunakan variabel

*KMeans.Converged*. Dalam *method Iterate()*, dideklarasikan 3 variabel *Hashtable()* yang bernama variabel *\_colourClusterAllocation*, variabel *\_pixelDataClusterAllocation*, dan variabel *\_clusterColours*, serta juga terdapat *method AllocateToCluster()* yang berfungsi untuk mengalokasikan nilai RGB setiap *pixel* ke dalam klaster terdekatnya, dan *method CalculateClusterCentroids()* yang berfungsi untuk mencari 2 warna *centroid* baru.

8. Pada *method AllocateToCluster()* terdapat proses menghitung jarak (deklarasasi variabel bertipekan *dictionary* bernama variabel *distance*). Sesuai dengan Persamaan 2.1, nilai RGB setiap *pixel* pada citra dihitung jaraknya, dengan code seperti berikut:

```
float d = (float)Math.Sqrt(
    (double)Math.Pow((c.Value.CentroidR - pd.Ch1), 2) +
    (double)Math.Pow((c.Value.CentroidG - pd.Ch2), 2) +
    (double)Math.Pow((c.Value.CentroidB - pd.Ch3), 2)
);
```

Dimana,

*c* : *\_currentCluster*, yang menyimpan nilai RGB setiap *centroid*

*pd* : *pixelData*, yang menyimpan nilai RGB dari *pixel* pada citra.

Hasil perhitungan jarak ke setiap *centroid* kemudian disimpan ke dalam variabel *distances*. Setiap nilai jarak warna *pixel* terhadap masing-masing *centroid* ini disimpan kembali di dalam variabel *List* yang bertipe *list*. Untuk melakukan alokasi klaster, data nilai jarak di dalam variabel

*list* ini dilakukan *sorting*, sehingga nilai jarak terkecil akan disimpan di dalam variabel *\_pixelDataClusterAllocation*.

9. Setelah melakukan alokasi klaster setiap *pixel* pada citra, langkah selanjutnya adalah melakukan pencarian kembali *centroid* baru pada *method CalculateClusterCentroids()*. Di dalam *method* ini, pertama kali dilakukan penyimpanan semua data RGB yang ada pada variabel *\_pixelDataClusterAllocation* ke dalam variabel *\_clusterColours* secara unik. Kemudian dilakukan perhitungan untuk mencari nilai rata2 masing-masing RGB dengan cara menjumlahkan nilai masing-masing RGB dari *centroid \_currentCluster* dengan total nilai RGB yang telah tersimpan di dalam variabel *array \_pixelDataClusterAllocation*, dan hasilnya disimpan kembali ke dalam nilai RGB *centroid* pada variabel *\_currentCluster* sehingga mengubah nilai RGB di dalam *array*-nya. Citra hasil dari proses segmentasi tahap ini kemudian disimpan ke dalam variabel *\_processedImage*.
10. Pada tahap ini dilakukan segmentasi warna. Yaitu dengan cara mengubah warna setiap *pixel* menjadi warna baru yang telah dialokasikan sebelumnya pada variabel *\_clusterColours*.
11. Setelah citra di segmentasi, langkah terakhir adalah melakukan cek status konvergen. Dengan mendeklarasikan dahulu variabel *match* yang bertipe *bool* dan diisi dengan nilai *true*. Kemudian melakukan pengecekan berupa perulangan *foreach* untuk setiap nilai RGB yang ada pada variabel *\_currentCluster* apakah terdapat perbedaan nilai dengan setiap nilai RGB pada variabel *\_previousCluster*. Apabila ditemukan perbedaan

nilai baik di salah satu nilai R, G, ataupun G, maka status konvergen variabel *match* diubah nilainya menjadi *false*, namun apabila sebaliknya maka tidak akan ada perubahan status konvergen dengan ini maka proses segmentasi algoritme *K-Means* sudah dianggap selesai. Nilai RGB *array* 2 klaster warna yang tersimpan di dalam variabel *array\_previousCluster* akan diubah nilainya menjadi nilai RGB *array* 2 klaster yang tersimpan pada variabel *\_currentCluster* hanya apabila status konvergen menjadi *false*.

12. Proses *method Iterate()* akan diulang kembali apabila nilai status konvergen yang disimpan pada variabel *match* masih berisi *false*. Namun apabila status konvergen sudah bernilai *true*, maka proses segmentasi berakhir dan akan dilanjutkan pada proses konversi hasil segmentasi citra *x-ray* menjadi biner.

Implementasi dari proses segmentasi menggunakan algoritme *K-Means* dalam bentuk *source code* ditampilkan pada Tabel 4.5.

**Tabel 4.5** Souce Code Segmentasi Citra X-Ray Paru-Paru

No.	Kode
1	private void button1_Click(object sender, EventArgs e)
2	{
3	Cursor.Current = Cursors.WaitCursor;
4	this.pictureBox38.Image = null;
5	this.pictureBox37.Image = null;
6	this.label4.Visible = true;
7	this.label7.Visible = true;
8	this.pictureBox2.Visible = true;

9	<code>this.label8.Visible = true;</code>
10	<code>this.pictureBox3.Visible = true;</code>
11	<code>this.label5.Visible = true;</code>
12	<code>this.pictureBox4.Visible = true;</code>
13	<code>//FILTER BACKGROUND</code>
14	<code>this.Segmentasi_Filter_Background();</code>
15	<code>//MULAI SEGMENTASI DENGAN K-MEANS</code>
16	<code>_kMeans = new KMeans((Bitmap)this.pictureBox8.Image, Convert.ToInt32(textBox1.Text), ImageProcessor.Colour.Types.RGB);</code>
17	<code>timer1.Enabled = true;</code>
18	<code>timer1.Start();</code>
19	<code>}</code>
20	<code>private void timer1_Tick(object sender, EventArgs e)</code>
21	<code>{</code>
22	<code>this.label4.Text = "K-Means : " + this.textBox1.Text.ToString() + " klaster.";</code>
23	<code>if (!_kMeans.Converged)</code>
24	<code>{</code>
25	<code>_kMeans.Iterate();</code>
26	<code>this.pictureBox2.Refresh();</code>
27	<code>this.pictureBox2.Image = _kMeans.ProcessedImage;</code>
28	<code>_count++;</code>
29	<code>this.label7.Text = "Iterasi: " + _count.ToString();</code>
30	<code>}</code>
31	<code>else</code>
32	<code>{</code>
33	<code>this.label7.Text = "Konvergen setelah " + _count.ToString() + " iterasi.";</code>
34	<code>_count = 0;</code>
35	<code>//SIMPAN GAMBAR HASIL SEGMENTASI</code>

36	<code>this.Save_Img_Segmentation();</code>
37	<code>//UBAH HASIL SEGMENTASI MENJADI BINARY</code>
38	<code>this.Segmentasi_To_Binary();</code>
39	<code>//INVERSI SEGMENTASI BINARY</code>
40	<code>this.Inversi_Segmentasi_Binary();</code>
41	<code>timer1.Enabled = false;</code>
42	<code>timer1.Stop(); }</code>

Tabel 4.5 menampilkan sebuah *method source code* untuk memulai proses segmentasi dan pemanggilan *method* algoritme *K-Means*, dengan data input  $k = 2$  untuk hasil akhir menjadi 2 kluster warna dan data citra *x-ray* yang telah di lakukan *filter* warna latar belakang kemudian melakukan *scanning* latar belakang untuk warna hitam saja. Tabel 4.6 menerangkan penjelasan dari masing-masing perintah setiap baris Tabel 4.5.

**Tabel 4.6** Penjelasan Fungsi Dari Setiap Perintah Tabel 4.5

Baris	Fungsi
1	Deklarasi <i>method</i> untuk tombol button1
1-12	Inisialisasi <i>picturebox</i> yang akan digunakan untuk menampilkan hasil proses segmentasi hingga hasil akhir segmentasinya.
14	Pemanggilan <i>method</i> <i>Segmentasi_Filter_Background</i>
16	Pemanggilan <i>method</i> <i>_kMeans</i> untuk melakukan proses segmentasi menggunakan algoritme <i>K-Means</i> untuk menentukan pusat klater awal dan klasterisasi warna awal.
17-18	Melakukan inisialiasi <i>timer1</i> , di mana pada <i>method</i> ini iterasi penentuan pusat kluster dan klasterisasi warna dilakukan.
20	Deklarasi <i>method</i> <i>timer1</i> untuk memulai iterasi penentuan pusat kluster dan klasterisasi warna.
23-43	Inisialisasi kondisi, apabila terjadi konvergen, di mana pusat kluster saat ini sama dengan pusat kluster pada iterasi sebelumnya, maka proses segmentasi dihentikan, apabila tidak sama maka akan ditentukan pusat klater baru dan klasterisasi warna kembali.

25	Melakukan iterasi kembali untuk menyimpan pusat kluster terbaru dan melakukan klasterisasi warna berdasarkan pusat kluster yang baru.
27	Menampilkan hasil klasterisasi pada iterasi saat ini pada <i>picturebox2</i> .
33	Menampilkan nilai total iterasi yang dilakukan selama proses segmentasi hingga ditemukan kondisi konvergen.
36	Memanggil <i>method Save_Img_Segmentation()</i> .
38	Memanggil <i>method Segmentasi_To_Binary()</i> untuk melakukan proses konversi citra x-ray menjadi biner dari hasil segmentasi.
40	Memanggil <i>method Inversi_Segmentasi_Binary()</i> untuk melakukan proses inversi citra x-ray hasil segmentasi.

Untuk *source code* dari *method kMeans.Iterate()* yang berfungsi untuk melakukan iterasi pencarian centroid baru, *method CalculateClusterCentroids()*, *method CheckConvergence()* terlampir pada bagian Lampiran nomor 4.

#### 4.3.3 Konversi Hasil Segmentasi Citra X-Ray Paru-Paru Menjadi Biner

Implementasi inversi citra *x-ray* hasil segmentasi diawali dengan proses *scanning* untuk diambil data warna (dalam *RGB*) pada citra *x-ray* oleh sistem. Kemudian dilakukan pengubahan warna hitam (*RGB*: 0, 0, 0) menjadi warna putih (*RGB*: 255, 255, 255) dan pengubahan warna putih (*RGB*: 255, 255, 255) menjadi warna hitam (*RGB*: 0, 0, 0). Hasil implementasi dari proses ini dalam bentuk *source code* ditampilkan pada Tabel 4.7.

**Tabel 4.7** *Souce Code* Konversi Citra X-Ray Hasil Segmentasi Menjadi Biner

No.	Kode
1	private void Segmentasi_To_Binary()
2	{
3	Bitmap bmp = new Bitmap(this.pictureBox2.Image);
4	int width = bmp.Width;
5	int height = bmp.Height;

6	int lowR = 255;
7	int lowG = 255;
8	int lowB = 255;
9	int curR = 0;
10	int curG = 0;
11	int curB = 0;
12	//Mencari warna yang ada setelah hasil segmentasi menggunakan algoritme K-Means
13	for (int y = 0; y < height; y++)
14	{
15	for (int x = 0; x < width; x++)
16	{
17	Color colour = bmp.GetPixel(x, y);
18	//ekstraksi nilai ARGB dari variabel colour
19	curR = colour.R;
20	curG = colour.G;
21	curB = colour.B;
22	if (curR < lowR && curG < lowG && curB < lowB)
23	{
24	lowR = curR;
25	lowG = curG;
26	lowB = curB;
27	}
28	}
29	}
30	//Mengubah Warna paling gelap jadi hitam dan warna paling terang jadi putih
31	for (int y = 0; y < height; y++)
32	{
33	for (int x = 0; x < width; x++)



34	{
35	Color colour = bmp.GetPixel(x, y);
36	//ekstraksi nilai ARGB dari variabel colour
37	int curA = colour.A;
38	curR = colour.R;
39	curG = colour.G;
40	curB = colour.B;
41	if (curR == lowR && curG == lowG && curB == lowB)
42	{
43	bmp.SetPixel(x, y, Color.FromArgb(curA, 0, 0, 0));
44	}
45	else
46	{
47	bmp.SetPixel(x, y, Color.FromArgb(curA, 255, 255, 255));
48	}
49	}
50	}
51	this.pictureBox3.Image = bmp;
52	this.Save_Img_Segmentasi_Binary();
53	}

Tabel 4.7 menampilkan sebuah *method source code* untuk melakukan konversi citra *x-ray* paru-paru hasil segmentasi menjadi citra biner. Sedangkan pada Tabel 4.8 menerangkan penjelasan dari masing-masing perintah setiap baris Tabel 4.7.

**Tabel 4.8** Penjelasan Fungsi Dari Setiap Perintah Tabel 4.7

Baris	Fungsi
1	Deklarasi <i>method</i> Segmentasi_To_Binary()

3	Mengambil citra yang telah di pilih dan ditampilkan pada <i>picturebox1</i> sebagai citra <i>inputan</i> untuk proses ini. Dan dikonversi menjadi <i>bitmap</i> .
4	Deklarasi variabel <i>width</i> untuk menentukan lebar <i>bitmap</i> .
5	Deklarasi variabel <i>height</i> untuk menentukan lebar <i>bitmap</i> .
6-8	Deklarasi variabel <i>lowR</i> , <i>lowG</i> , dan <i>lowB</i> untuk menyimpan nilai warna <i>RGB</i> pada citra hasil segmentasi yang paling gelap.
9-11	Deklarasi variabel <i>curR</i> , <i>curG</i> , dan <i>curB</i> untuk menyimpan nilai warna <i>RGB</i> pada setiap kordinat ketika dilakukan <i>scanning</i> citra <i>x-ray</i> paru-paru dari hasil segmentasi.
13	Inisialisasi looping kordinat <i>bitmap</i> dimulai dari <i>axis y</i> pada titik ke 0 sampai ke titik 255.
15	Inisialisasi looping kordinat <i>bitmap</i> dimulai dari <i>axis x</i> pada titik ke 0 sampai ke titik 255.
17-21	Pengambilan warna pada kordinat saat ini, yang diambil adalah nilai <i>RGB</i> dan disimpan pada variabel <i>curR</i> , <i>curG</i> , dan <i>curB</i> .
22-27	Inisialisasi kondisi, apabila nilai <i>RGB(curR, curG, curB)</i> kordinat saat ini kurang dari <i>RGB(lowR, lowG, lowB)</i> , maka menyimpan nilai <i>RGB(curR, curG, curB)</i> ke dalam variabel <i>RGB (lowR, lowG, lowB)</i> .
31	Inisialisasi looping kordinat <i>bitmap</i> dimulai dari <i>axis y</i> pada titik ke 0 sampai ke titik 255.
33	Inisialisasi looping kordinat <i>bitmap</i> dimulai dari <i>axis x</i> pada titik ke 0 sampai ke titik 255.
35-40	Pengambilan warna pada kordinat saat ini, yang diambil adalah nilai <i>RGB</i> dan disimpan pada variabel <i>curR</i> , <i>curG</i> , dan <i>curB</i> .
41-48	Inisialisasi kondisi, apabila nilai <i>RGB(curR, curG, curB)</i> kordinat saat ini sama dengan nilai <i>RGB</i> pada variabel <i>lowR</i> , <i>lowG</i> , dan <i>lowB</i> . Maka warna pada kordinat saat ini diubah menjadi hitam ( <i>RGB: 0, 0, 0</i> ), namun apabila tidak maka diubah menjadi warna putih ( <i>RGB: 255, 255, 255</i> ).
51	Tampilkan hasil citra dari proses ini pada <i>pictureBox3</i> .
52	Simpan hasil citra dari proses ini di dalam sistem.

#### 4.3.4 Inversi Hasil Segmentasi Citra X-Ray Paru-Paru

Implementasi inversi citra *x-ray* hasil segmentasi diawali dengan proses *scanning* untuk diambil data warna (dalam *RGB*) pada citra *x-ray* oleh sistem. Kemudian dilakukan pengubahan warna hitam (*RGB*: 0, 0, 0) menjadi warna putih (*RGB*: 255, 255, 255) dan pengubahan warna putih (*RGB*: 255, 255, 255) menjadi warna hitam (*RGB*: 0, 0, 0). Kemudian hasilnya ditampilkan di dalam aplikasi. Implementasi dari proses ini dalam bentuk *source code* ditampilkan pada Tabel 4.9.

**Tabel 4.9** Souce Code Inversi Citra X-Ray Paru-Paru Hasil Segmentasi

No.	Kode
1	private void Inversi_Segmentasi_Binary()
2	{
3	Bitmap bmp = new Bitmap(this.pictureBox3.Image);
4	int width = bmp.Width;
5	int height = bmp.Height;
6	for (int y = 0; y < height; y++)
7	{
8	for (int x = 0; x < width; x++)
9	{
10	Color colour = bmp.GetPixel(x, y);
11	//ekstraksi nilai ARGB dari variabel colour
12	int alpha = colour.A;
13	int red = colour.R;
14	int green = colour.G;
15	int blue = colour.B;
16	if (red == 0 && green == 0 && blue == 0)
17	{

18	<code>bmp.SetPixel(x, y, Color.FromArgb(alpha, 255, 255, 255));</code>
19	<code>}</code>
20	<code>else if (red == 255 &amp;&amp; green == 255 &amp;&amp; blue == 255)</code>
21	<code>{</code>
22	<code>bmp.SetPixel(x, y, Color.FromArgb(alpha, 0, 0, 0));</code>
23	<code>}</code>
24	<code>}</code>
25	<code>}</code>
26	<code>this.pictureBox4.Image = bmp;</code>
27	<code>this.Save_Img_Segmentasi_Inversi();</code>
28	<code>Cursor.Current = Cursors.Default;</code>
29	<code>}</code>

Tabel 4.9 menampilkan sebuah *method source code* untuk melakukan inversi citra *x-ray* paru-paru hasil segmentasi. Sedangkan pada Tabel 4.10 menerangkan penjelasan dari masing-masing perintah setiap baris Tabel 4.9.

**Tabel 4.10** Penjelasan Fungsi Dari Setiap Perintah Tabel 4.9

Baris	Fungsi
1	Deklarasi <i>method</i> <code>Inversi_Segmentasi_Binary()</code>
3	Mengambil citra yang telah di pilih dan ditampilkan pada <i>picturebox1</i> sebagai citra <i>inputan</i> untuk proses ini. Dan dikonversi menjadi <i>bitmap</i> .
4	Deklarasi variabel <code>witdh</code> untuk menentukan lebar <i>bitmap</i> .
5	Deklarasi variabel <code>height</code> untuk menentukan lebar <i>bitmap</i> .
6	Inisialisasi looping kordinat <i>bitmap</i> dimulai dari <i>axis y</i> pada titik ke 0 sampai ke titik 255.
8	Inisialisasi looping kordinat <i>bitmap</i> dimulai dari <i>axis x</i> pada titik ke 0 sampai ke titik 255.
12-15	Pengambilan warna pada kordinat saat ini, yang diambil adalah nilai <i>aRGB</i> dan disimpan pada variabel <code>alpha</code> , <code>red</code> , <code>green</code> , dan <code>blue</code> .

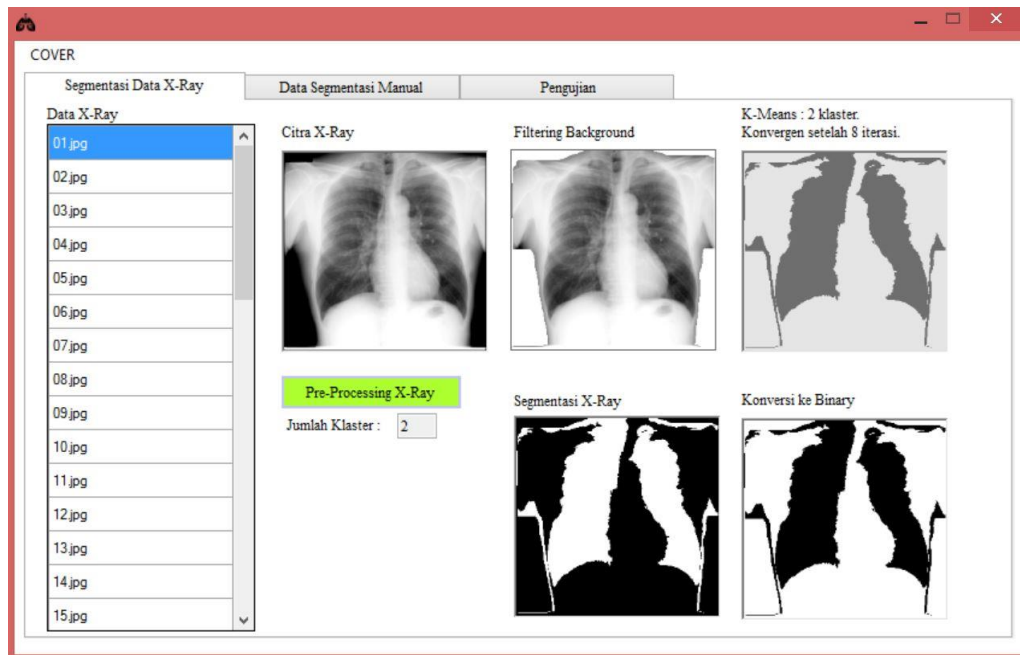
16-23	Inisialisasi kondisi, apabila nilai RGB(red, green, blue) kordinat saat ini sama dengan RGB(0, 0, 0), maka diubah warnanya menjadi RGB(255, 255, 255) sedangkan apabila nilai RGB(red, green, blue) kordinat saat ini sama dengan RGB(255, 255, 255), maka diubah warnanya menjadi RGB(0, 0, 0).
26	Tampilkan hasil citra dari proses ini pada <i>pictureBox4</i> .
27	Simpan hasil citra dari proses ini di dalam sistem.

#### 4.4 Implementasi Antarmuka

Pada bagian implementasi antarmuka ini menjelaskan tampilan antarmuka dari aplikasi yang telah dibangun pada penelitian ini berdasarkan hasil perancangan yang telah dilakukan sesuai pada Bab 3.6. Antarmuka yang telah dibangun terdiri dari 3 tampilan utama.

##### 4.4.1 Implementasi Tab Segmentasi Data *X-Ray*

Pada tampilan Tab Segmentasi Data *X-Ray* terdapat berbagai fitur untuk melakukan segmentasi pada citra *x-ray* paru-paru. Termasuk tabel data asli citra *x-ray* paru-paru yang tersimpan di dalam sistem. Setiap tahapan dari rangkaian proses segmentasi ditampilkan dalam kotak gambar tersendiri. Dimulai dari tombol untuk memulai segmentasi citra *x-ray* paru-paru, tahap penghapusan warna hitam pada latar belakang menjadi putih, segmentasi citra *x-ray* paru-paru menggunakan algoritme *K-Means* lengkap dengan label jumlah iterasi yang terjadi, kemudian dilanjutkan dengan konversi citra hasil segmentasi, dan terakhir adalah inversi citra hasil segmentasi. Gambar 4.2 menampilkan antarmuka dari Tab Segmentasi Data *X-Ray* yang telah di bangun di dalam sistem pada penelitian ini.



**Gambar 4.2** Tampilan Tab Segmentasi Data X-Ray

#### 4.4.2 Implementasi Tab Data Segmentasi Manual

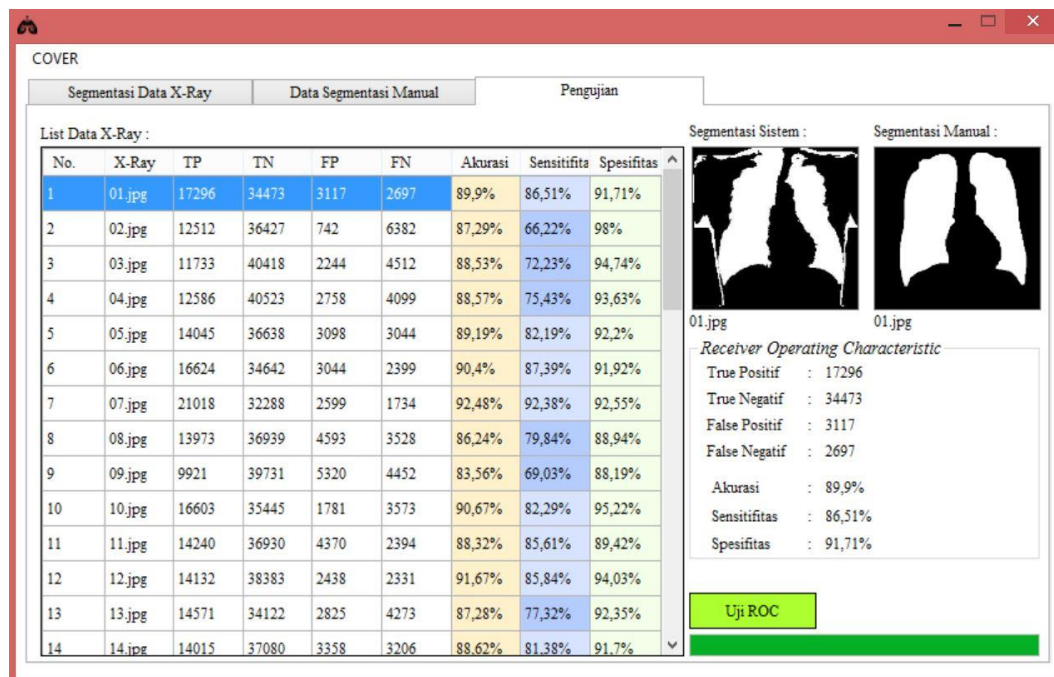
Pada tampilan Tab Segmentasi Manual ditampilkan data citra *x-ray* paru-paru yang telah disegmentasi secara manual menggunakan fitur *Magic Wand* pada aplikasi *Adobe Photoshop*. Terdapat sebanyak 45 data citra yang ditampilkan, diurutkan nomor pendataan citra dari kiri ke kanan dengan penampilan sebanyak 5 citra setiap barisnya. Gambar 4.3 menampilkan antarmuka dari Tab Segmentasi Manual yang telah di bangun di dalam sistem pada penelitian ini.



**Gambar 4.3** Tampilan Tab Data Segmentasi Manual

#### 4.4.3 Implementasi Tab Pengujian

Pada tampilan Tab Pengujian terdapat tampilan untuk melakukan pengujian pada hasil segmentasi pada citra *x-ray* paru-paru dengan menggunakan algoritme *K-Means*. Pada bagian kiri menampilkan sebuah table data nilai TP, TN FP, dan FN untuk setiap citra *x-ray*. Pada bagian kanan atas terdapat 2 buah gambar sebagai perbandingan antara citra yang telah disegmentasi oleh sistem dan citra yang telah disegmentasi secara manual menggunakan aplikasi *Adobe Photoshop*. Terdapat tombol “Uji ROC” pada bagian kanan bawah untuk memulai menghitung nilai akurasi, spesifitas, dan sensitifitas dari sebanyak 45 citra hasil segmentasi oleh sistem. Gambar 4.4 menampilkan antarmuka dari Tab Pengujian yang telah dibangun di dalam sistem pada penelitian ini.



**Gambar 4.4** Tampilan Tab Pengujian

#### 4.5 Pengujian Dan Pembahasan

Metode pengujian yang digunakan untuk mencari nilai evaluasi dari penelitian ini adalah metode *Receiver Operating Characteristics*. Metode ini menggunakan 4 parameter untuk kemudian digunakan dalam perhitungan akurasi, sensitifitas, dan spesifitasnya yang mana akan menentukan seberapa besar kesesuaian dari hasil segmentasi oleh sistem.

Untuk mendapatkan nilai TP, TN, FP, dan FN, dilakukan perbandingan masing-masing citra *x-ray* hasil segmentasi oleh sistem dengan citra *x-ray* hasil segmentasi secara manual sesuai yang telah dirancang pada Bab 3.7. Metode pengujian ini diterapkan pada citra hasil segmentasi oleh sistem secara utuh tanpa memisah antara paru-paru kanan dengan paru-paru kiri tersendiri.

Perhitungan akurasi, sensitifitas, dan spesifitas dari 45 citra *x-ray* hasil segmentasi oleh sistem menggunakan Persamaan 3.1 untuk akurasi, Persamaan 3.2



untuk perhitungan sensitifitas, dan Persamaan 3.3 untuk perhitungan spesifitas. Tabel 4.11 menampilkan kumpulan data dari keempat parameter (TP, TN, FP, dan FN) dan nilai akurasi, sensitifitas, serta spesifitas dari 45 citra *x-ray* hasil dari segmentasi citra oleh sistem yang dibandingkan dengan citra hasil dari segmentasi secara manual.

**Tabel 4.11.** Hasil Pengujian *Receiver Operating Characteristic*

No.	X-Ray	TP	TN	FP	FN	Akurasi	Sensitifitas	Spesifitas
1	01.jpg	17296	34473	3117	2697	89,9%	86,51%	91,71%
2	02.jpg	12512	36427	742	6382	87,29%	66,22%	98%
3	03.jpg	11733	40418	2244	4512	88,53%	72,23%	94,74%
4	04.jpg	12586	40523	2758	4099	88,57%	75,43%	93,63%
5	05.jpg	14045	36638	3098	3044	89,19%	82,19%	92,2%
6	06.jpg	16624	34642	3044	2399	90,4%	87,39%	91,92%
7	07.jpg	21018	32288	2599	1734	92,48%	92,38%	92,55%
8	08.jpg	13973	36939	4593	3528	86,24%	79,84%	88,94%
9	09.jpg	9921	39731	5320	4452	83,56%	69,03%	88,19%
10	10.jpg	16731	35609	1696	3309	91,27%	83,49%	95,45%
11	11.jpg	14240	36930	4370	2394	88,32%	85,61%	89,42%
12	12.jpg	14132	38383	2438	2331	91,67%	85,84%	94,03%
13	13.jpg	14571	34122	2825	4273	87,28%	77,32%	92,35%
14	14.jpg	14015	37080	3358	3206	88,62%	81,38%	91,7%
15	15.jpg	14974	34528	1328	5794	87,42%	72,1%	96,3%
16	16.jpg	13626	39576	4719	1378	89,72%	90,82%	89,35%
17	17.jpg	24086	29571	1370	2762	92,85%	89,71%	95,57%
18	18.jpg	16174	37160	3426	1445	91,63%	91,8%	91,56%
19	19.jpg	16047	36890	3100	1861	91,43%	89,61%	92,25%
20	20.jpg	18419	36740	1545	1784	94,31%	91,17%	95,96%
21	21.jpg	15770	35877	2850	3059	89,73%	83,75%	92,64%
22	22.jpg	16887	33232	2742	3649	88,69%	82,23%	92,38%
23	23.jpg	13393	38667	3211	2621	89,93%	83,63%	92,33%

24	24.jpg	12925	37929	2691	3526	89,11%	78,57%	93,38%
25	25.jpg	17558	34259	3041	2516	90,31%	87,47%	91,85%
26	26.jpg	17842	33661	2261	2662	91,28%	87,02%	93,71%
27	27.jpg	11082	41065	2928	4000	88,27%	73,48%	93,34%
28	28.jpg	14638	35187	2184	4608	88%	76,06%	94,16%
29	29.jpg	15419	36405	2787	2363	90,96%	86,71%	92,89%
30	30.jpg	15285	37401	443	3507	93,03%	81,34%	98,83%
31	31.jpg	19859	33685	2179	1965	92,82%	91%	93,92%
32	32.jpg	12131	39318	3610	2987	88,63%	80,24%	91,59%
33	33.jpg	14015	37723	1714	3333	91,11%	80,79%	95,65%
34	34.jpg	14838	36220	2727	3301	89,44%	81,8%	93%
35	35.jpg	18369	32096	3778	2534	88,88%	87,88%	89,47%
36	36.jpg	17126	37221	807	3548	92,58%	82,84%	97,88%
37	37.jpg	9514	41433	4054	4191	86,07%	69,42%	91,09%
38	38.jpg	13984	36978	3031	3410	88,78%	80,4%	92,42%
39	39.jpg	18419	34118	3848	1550	90,68%	92,24%	89,86%
40	40.jpg	17156	36095	2109	2207	92,5%	88,6%	94,48%
41	41.jpg	15883	36165	3248	2021	90,81%	88,71%	91,76%
42	42.jpg	9764	40325	2927	5148	86,12%	65,48%	93,23%
43	43.jpg	17582	36420	3066	1489	92,22%	92,19%	92,24%
44	44.jpg	13336	37093	4352	2651	87,81%	83,42%	89,5%
45	45.jpg	16173	35040	2598	2986	90,17%	84,41%	93,1%
<b>Rata-rata</b>		<b>15237</b>	<b>36495</b>	<b>2819</b>	<b>3094</b>	<b>89,75%</b>	<b>82,66%</b>	<b>92,90%</b>

Berdasarkan Tabel 4.11, didapatkan nilai rata-rata akurasi sebesar 89,75%, nilai rata-rata sensitifitas sebesar 82,66%, dan nilai rata-rata spesifitas sebesar 92,90%. Ketiga nilai evaluasi ini dinilai sangat baik apabila dibandingkan dengan hasil penelitian dengan objek serupa oleh Ainatur Mardhiyah, et al. (2011) yang mendapat nilai evaluasi sensitifitas sebesar 62,05% untuk paru-paru kiri dan sebesar 63,71% untuk paru-paru kanan.

Apabila nilai evaluasi dari penelitian ini dibandingkan dengan hasil penelitian dengan objek yang serupa oleh I Made Oka Widyantara, et al. (2015) memiliki nilai evaluasi yang lebih rendah. Dimana pada penelitian tersebut didapatkan nilai evaluasi akurasi sebesar 92,87%, sensitifitas sebesar 93,44%, dan spesifitas sebesar 89,38% untuk paru-paru kiri dan nilai akurasi sebesar 92,47%, sensitifitas sebesar 93,89%, dan spesifitas sebesar 85,89% untuk paru-paru kanan.

Berdasarkan perbandingan hasil pengujian dengan kedua penelitian terdahulu dengan objek yang serupa tersebut, penelitian ini memiliki peluang untuk dilakukan pengembangan lebih lanjut. Sehingga penelitian ini khususnya pada penelitian segmentasi citra *xray thorax* tersebut menjadi lebih sempurna dan bermanfaat bagi kalangan dokter maupun masyarakat.

Segmentasi paru-paru pada citra *x-ray thorax* ini mengklusterisasi warna citra menjadi dua dengan menggunakan algoritme K-Means. Untuk dapat melakukan klusterisasi pada citra *x-ray* dibutuhkan klusterisasi warna yang berbeda pada organ paru-paru dan organ yang lainnya. al-qur'an telah menjelaskan bahwa Allah menciptakan langit dan bumi dengan bahasa dan warna kulit yang berbeda-beda, hal tersebut dijelaskan pada surah (QS. Ar-rum: 22) yang berbunyi:

وَمِنْ آيَاتِهِ خَلْقُ السَّمَوَاتِ وَالْأَرْضِ وَأَخْتَلَفُ اللَّسَانَاتِ وَاللَّوْنَاتِ إِنَّ فِي ذَلِكَ لَآيَاتٍ  
لِّلْعَالَمِينَ ٢٢

Artinya: *Dan di antara tanda-tanda kekuasaan-Nya ialah menciptakan langit dan bumi dan berlain-lainan bahasamu dan warnamu. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda bagi orang-orang yang mengetahui (Al-Qur'an Terjemah, 2006).*

Dari ayat tersebut dapat dijadikan pijakan skripsi yang peneliti angkat. Merujuk pada kalimat *وَاللَّوْنَاتِ* yang berarti warnamu, yang dimaksud dengan

*warnamu* adalah arti luas dari kalimat pada ayat tersebut. Warna sendiri memiliki banyak macam-macamnya, bisa warna yang dimaksud adalah warna kulit ataupun warna yang lainnya. Berdasarkan arti kalimat ayat tersebut penulis mengkaitkan dengan penelitian yang penulis kerjakan dimana bahwa dengan melakukan klasterisasi dua warna pada citra *x-ray* paru-paru menjadi salah satu indikator dalam pendeteksian penyakit pada organ paru-paru.

Seperti yang sudah penulis kerjakan pada penelitian diatas, untuk mendapatkan hasil segmentasi yang diinginkan maka data foto *x-ray thorax* harus melalui tahap *pre-processing* yang selanjutnya hasil dari *pre-processing* tersebut dilakukan klasterisasi menggunakan algoritme *K-Means*, pada tahap tersebut nilai *K* atau nilai jumlah klaster yang digunakan adalah 2, dimana nilai tersebut mewakili banyaknya warna yang menjadi tolak ukur atau patokan yang akan dilakukan pada proses segmentasi data citra *x-ray thorax* tersebut.

Pada penelitian tersebut, 2 warna yang dimaksudkan adalah warna hitam dan warna putih. Berdasarkan isi dari kandungan ayat diatas, penulis berusaha untuk mengaitkan dari perbedaan warna yang diciptakan oleh Allah SWT. Pada penelitian ini, penulis menggunakan warna hitam dan putih tersebut sebagai pembeda antara *background* dengan objek paru-paru yang telah disegmentasi.

## **BAB V PENUTUP**

### **5.1 Kesimpulan**

Berikut ini adalah beberapa hal yang dapat disimpulkan dari penelitian ini adalah:

1. Algoritme *K-Means* telah berhasil diimplementasikan menggunakan bahasa pemrograman C# dengan berbagai fitur didalamnya. Fitur yang berhasil diimplementasikan adalah segmentasi citra *x-ray* menggunakan algoritme *K-Means*, data segmentasi citra *x-ray* secara manual, dan pengujian dengan metode *Receiver Operating Characteristics* pada algoritme *K-Means* yang diterapkan pada sistem.
2. Pengujian pada sistem telah dilakukan dengan metode *Receiver Operating Characteristics* pada perbandingan antara 45 citra *x-ray* hasil segmentasi oleh sistem dengan 45 segmentasi secara manual menggunakan fitur *Magic Wand* pada aplikasi *Adobe Photoshop*. Hasil akurasi yang didapatkan nilai rata-rata akurasi sebesar 89,75%, nilai rata-rata sensitifitas sebesar 82,66%, dan nilai rata-rata spesifitas sebesar 92,90%. Berdasarkan nilai evaluasi yang didapatkan dari penelitian ini, masih dapat dilakukan pengembangan lebih lanjut.

### **5.2 Saran**

Berikut ini adalah beberapa saran yang diajukan untuk penelitian pengembangan lebih lanjut.

1. Menggunakan metode klustering yg lain yang mampu mendeteksi area paru-paru dengan lebih optimal sehingga hasil segmentasinya benar-benar hanya menampilkan area paru-paru saja.
2. Menerapkan sebuah metode optimasi untuk mendeteksi garis batas tepi dari citra *x-ray* paru-paru sehingga area paru-paru yang tersegmentasi hasilnya lebih baik.

## DAFTAR PUSTAKA

- Agus Tommy Adi Prawira Kusuma, I Made Oka Widyantara, Ni Made Ary Esta Dewi Wirasturi, 2015, *Preprocessing pada Segmentasi Citra Paru-Paru dan Jantung Menggunakan Anisotropic Diffusion Filter*, Universitas Udayana, Bali.
- Ainatul Mardhiyah, Agus Harjoko, 2011, *Metode Segmentasi Paru-Paru dan Jantung Pada Citra X-Ray Thorax*, Indonesian Journal of Electronics and Instrumentation Systems Vol 1, No 2 (2011): IJEIS pp. 35-44.
- Akhilesh Kumar Yadav, Divya Tomar, Sonali Agarwal, 2013, *Clustering of Lung Cancer Data Using Foggy K-Means*, Indian Institute of Information Technology, Allahabad, India.
- Amin Haryadi, Linda Salma Anggraeni, A. Syahirul Fatkhurrahman, 2015, *Aplikasi Segmentasi Paru-Paru*, Teknik Informatika Universitas Islam Negeri Malik Ibrahim Malang, Malang.
- Baratloo A, Hosseini M, Negida A, El Ashal G. 2015. *Part 1: Simple Definition and Calculation of Accuracy, Sensitivity and Specificity. Emergency. 3(2):* pp. 48-49.
- C. Gabriel, S. Gabriel, dan E. Corthout, 1996, *The Dielectric Properties of Biological Tissues*, King's College, London, United Kingdom.
- Evelyn CP, 2009, *Anatomi dan Fisiologi untuk Paramedis*, Jakarta: PT Gramedia.
- Ferry Suyanto, 2008, *Aplikasi Radiasi Sinar-X Di Bidang Kedokteran untuk Menunjang Kesehatan Masyarakat Kawasan Puspitek Serpong*, Tangerang: Pusat Rekayasa Perangkat Nuklir-BATAN.
- Imran Farced Nizami, Saad UI Hasan, Ibrahim Tariq Javed, 2014, *A Wavelet Frames + K-means based Automatic Method for Lung Area Segmentation in Multiple Slices of CT Scan*, Bahria University, Islamabad, Pakistan.
- Isi.uu.nl/Research/Databases/SCR/. SCR database: Segmentation in Chest Radiographs Introduction. Diakses pada 13 Agustus 2016.
- Mutiara Hadits. "Setiap Penyakit Ada Obatnya." <https://www.mutiarahadits.com/82/66/76/setiap-penyakit-ada-obatnya.htm> (diakses pada tanggal 9 September 2017)
- P. B Sangamithraa dan S Govindaraju, 2016, *Lung Tumour Detection and Classification Using EK-Mean Clustering*, Kumaraguru College of Technology, Coimbatore, India.
- Reni Rahmadewi, Rahmadi Kurnia, 2016, *Klasifikasi Penyakit Paru Berdasarkan Citra Rontgen Dengan Metoda Segmentasi SOBEL*, Universitas Andalas, Padang.
- Richard S. Snell, 2006, *Clinical Anatomy by Systems, Volume 20, Issue 2* Maret 2007 pp. 223–224.





- Sabar Setiawidayat, Suci Imani Putri, *Filtering Data Diskrit Elektrokardiogram Untuk Penentuan PQRST Dalam Satu Siklus*, Universitas Widyagama, Malang.
- Sherwood L, 2001, *Fisiologi Manusia; dari Sel ke Sistem*, Edisi 2, Jakarta: EGC.
- Sutoyo, T, dkk. 2009, *Teori Pengolahan Citra Digital*, Penerbit Andi, Yogyakarta hal 9- 27.
- Tortora GJ, Derrickson B, 2012, *Principles of Anatomy and Physiology*. Edisi ke-13. New Jersey: John Wiley & Sons. pp. 918-66.
- TafsiQ. "Hadits Muslim Nomor 4084." <https://tafsirq.com/hadits/muslim/4084> (diakses pada tanggal 10 September 2017)

## LAMPIRAN







1. Data asli *x-ray* paru-paru yang digunakan dalam penelitian ini terdapat 45 gambar.







Tabel A1 menampilkan daftar data asli *x-ray* paru-paru yang disimpan dalam sistem untuk digunakan dalam implementasinya.







**Tabel A1.** Daftar Data Asli X-Ray Paru-Paru







No.	Nama <i>File</i> Gambar	Gambar	Ukuran ( <i>pixels</i> )
1	01.jpg		256 x 256
2	02.jpg		256 x 256
3	03.jpg		256 x 256
4	04.jpg		256 x 256















5	05.jpg		256 x 256
6	06.jpg		256 x 256
7	07.jpg		256 x 256
8	08.jpg		256 x 256
9	09.jpg		256 x 256
10	10.jpg		256 x 256






11	11.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is centered and shows clear lung fields.	256 x 256
12	12.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is centered and shows clear lung fields.	256 x 256
13	13.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is centered and shows clear lung fields.	256 x 256
14	14.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is centered and shows clear lung fields.	256 x 256
15	15.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is centered and shows clear lung fields.	256 x 256
16	16.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is centered and shows clear lung fields.	256 x 256

17	17.jpg	 A frontal chest X-ray showing clear lung fields, a normal heart size, and a visible diaphragm.	256 x 256
18	18.jpg	 A frontal chest X-ray showing clear lung fields, a normal heart size, and a visible diaphragm.	256 x 256
19	19.jpg	 A frontal chest X-ray showing clear lung fields, a normal heart size, and a visible diaphragm.	256 x 256
20	20.jpg	 A frontal chest X-ray showing clear lung fields, a normal heart size, and a visible diaphragm.	256 x 256
21	21.jpg	 A frontal chest X-ray showing clear lung fields, a normal heart size, and a visible diaphragm.	256 x 256
22	22.jpg	 A frontal chest X-ray showing clear lung fields, a normal heart size, and a visible diaphragm.	256 x 256

23	23.jpg		256 x 256
24	24.jpg		256 x 256
25	25.jpg		256 x 256
26	26.jpg		256 x 256
27	27.jpg		256 x 256
28	28.jpg		256 x 256

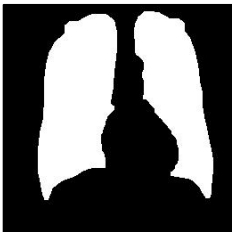

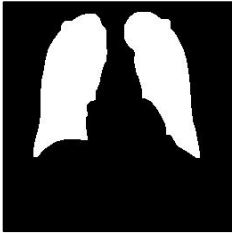

29	29.jpg		256 x 256
30	30.jpg		256 x 256
31	31.jpg		256 x 256
32	32.jpg		256 x 256
33	33.jpg		256 x 256
34	34.jpg		256 x 256

35	35.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is in grayscale and appears to be a standard diagnostic view.	256 x 256
36	36.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is in grayscale and appears to be a standard diagnostic view.	256 x 256
37	37.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is in grayscale and appears to be a standard diagnostic view.	256 x 256
38	38.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is in grayscale and appears to be a standard diagnostic view.	256 x 256
39	39.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is in grayscale and appears to be a standard diagnostic view.	256 x 256
40	40.jpg	 A frontal chest X-ray showing the lungs, heart, and diaphragm. The image is in grayscale and appears to be a standard diagnostic view.	256 x 256

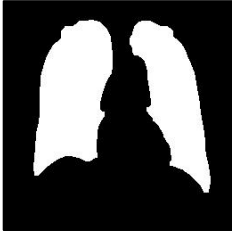
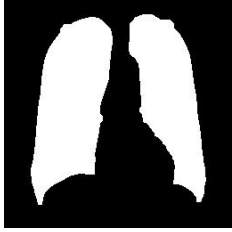
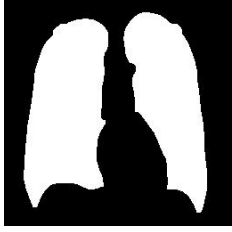
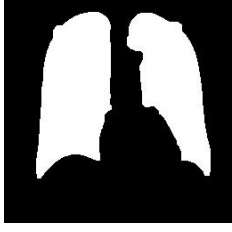
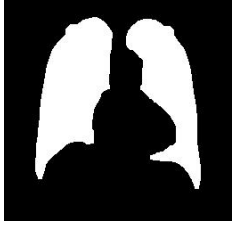
41	41.jpg		256 x 256
42	42.jpg		256 x 256
43	43.jpg		256 x 256
44	44.jpg		256 x 256
45	45.jpg		256 x 256



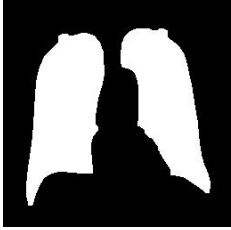



2. Data hasil segmentasi gambar *x-ray* paru-paru secara manual menggunakan teknik pemotongan dengan *Magic Wand* pada aplikasi *Adobe Photoshop*. Kemudian dilakukan inversi gambar potongan menjadi gambar dengan skala biner. Terdapat 45 data gambar *x-ray* paru-paru yang telah dilakukan segmentasi secara manual. Tabel A2 menampilkan daftar data segmentasi citra *x-ray* paru-paru secara manual.



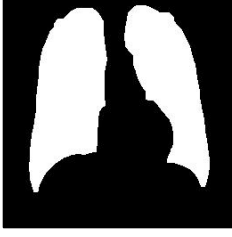
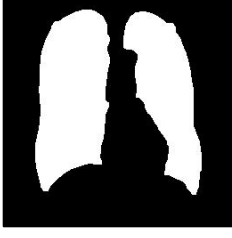
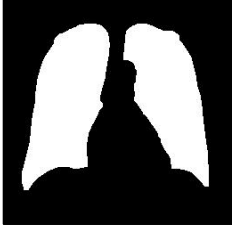

**Tabel A2.** Daftar Data Segmentasi Secara Manual

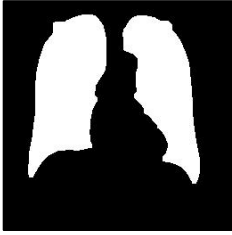
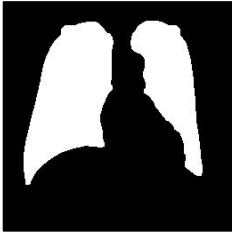
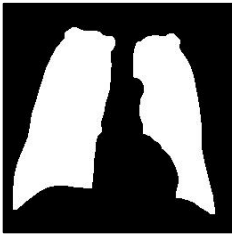


No.	Nama <i>File</i> Gambar	Gambar	Ukuran ( <i>pixels</i> )
1	01.jpg		256 x 256
2	02.jpg		256 x 256
3	03.jpg		256 x 256
4	04.jpg		256 x 256


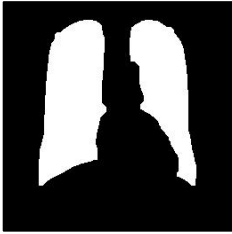
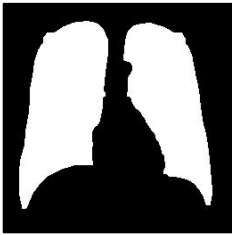
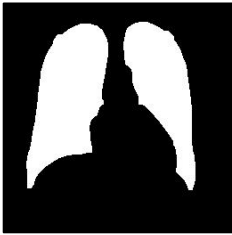

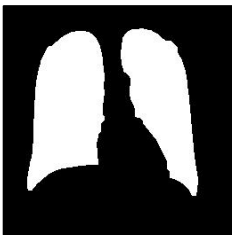




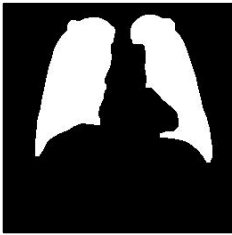

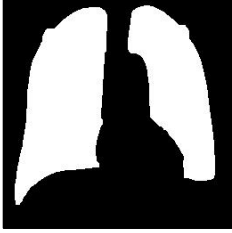

5	05.jpg		256 x 256
6	06.jpg		256 x 256
7	07.jpg		256 x 256
8	08.jpg		256 x 256
9	09.jpg		256 x 256
10	10.jpg		256 x 256


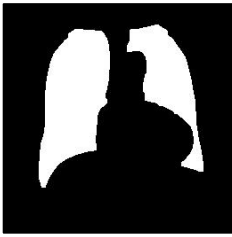
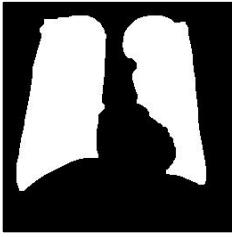


11	11.jpg		256 x 256
12	12.jpg		256 x 256
13	13.jpg		256 x 256
14	14.jpg		256 x 256
15	15.jpg		256 x 256
16	16.jpg		256 x 256

17	17.jpg		256 x 256
18	18.jpg		256 x 256
19	19.jpg		256 x 256
20	20.jpg		256 x 256
21	21.jpg		256 x 256
22	22.jpg		256 x 256

23	23.jpg		256 x 256
24	24.jpg		256 x 256
25	25.jpg		256 x 256
26	26.jpg		256 x 256
27	27.jpg		256 x 256
28	28.jpg		256 x 256

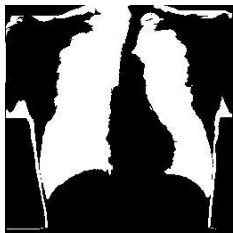
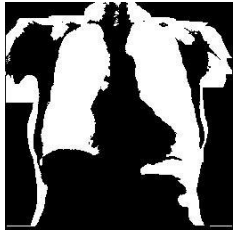
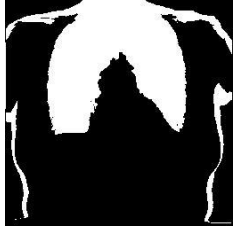
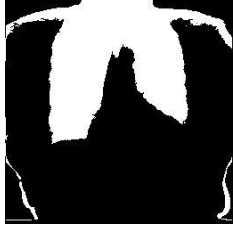
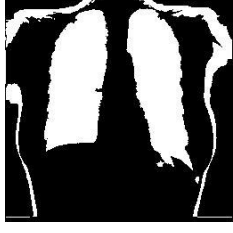
29	29.jpg		256 x 256
30	30.jpg		256 x 256
31	31.jpg		256 x 256
32	32.jpg		256 x 256
33	33.jpg		256 x 256
34	34.jpg		256 x 256

35	35.jpg		256 x 256
36	36.jpg		256 x 256
37	37.jpg		256 x 256
38	38.jpg		256 x 256
39	39.jpg		256 x 256
40	40.jpg		256 x 256

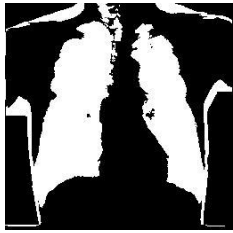
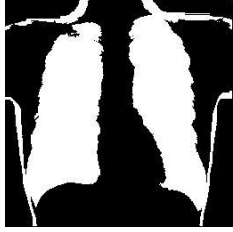
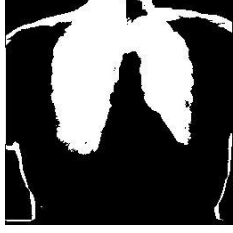
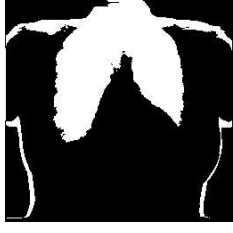

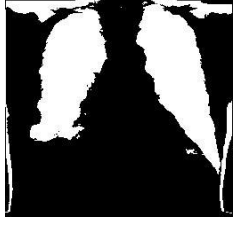
41	41.jpg		256 x 256
42	42.jpg		256 x 256
43	43.jpg		256 x 256
44	44.jpg		256 x 256
45	45.jpg		256 x 256

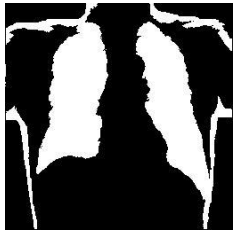
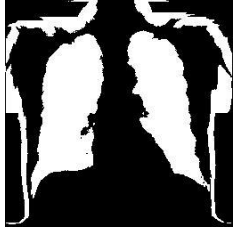
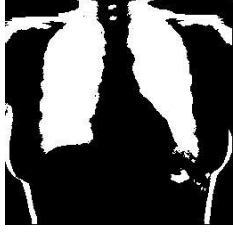
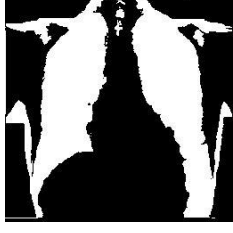

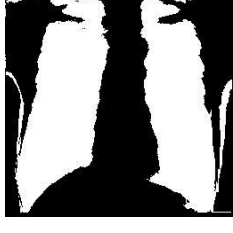
3. Data segmentasi citra *x-ray* paru-paru yang dihasilkan oleh sistem pada penelitian ini terdapat 45 gambar. Tabel A3 menampilkan daftar data segmentasi citra *x-ray* paru-paru yang dihasilkan oleh sistem.

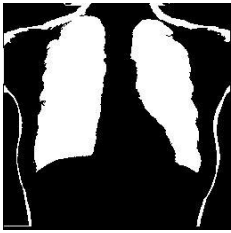
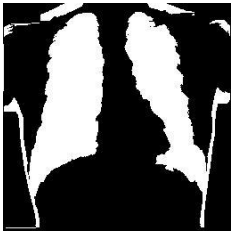
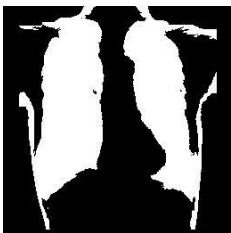
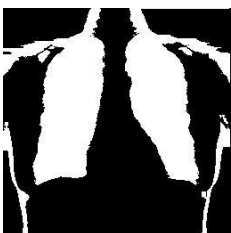
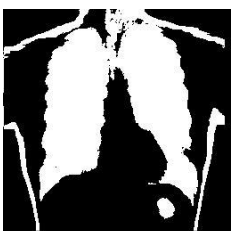
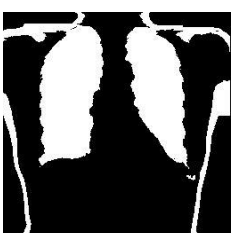
**Tabel A3.** Daftar Data Hasil Segmentasi Citra *X-Ray* Oleh Sistem

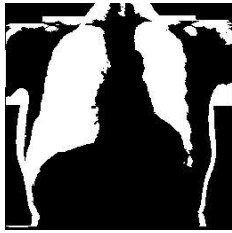
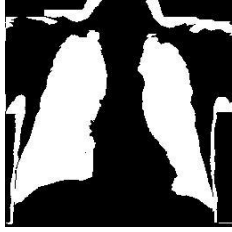
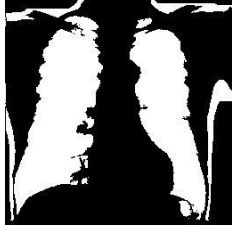
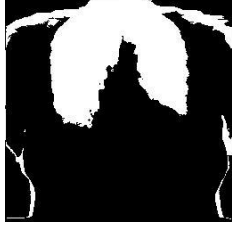
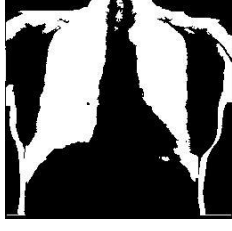
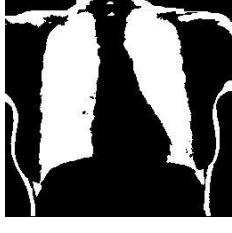
No.	Nama <i>File</i> Gambar	Gambar	Ukuran ( <i>pixels</i> )
1	01.jpg		256 x 256
2	02.jpg		256 x 256
3	03.jpg		256 x 256
4	04.jpg		256 x 256
5	05.jpg		256 x 256

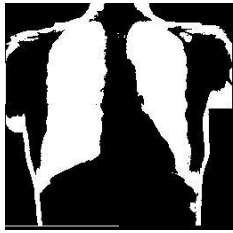
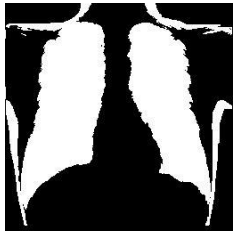
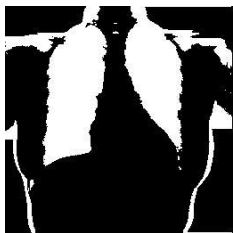
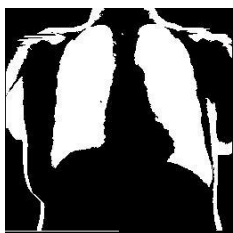
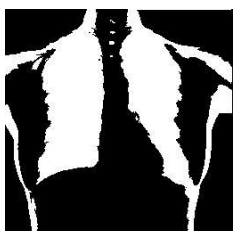



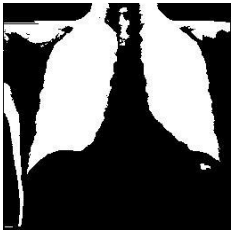
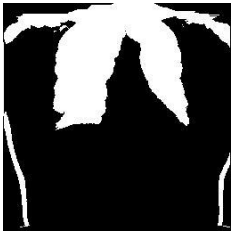
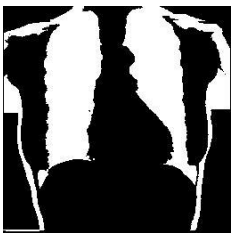
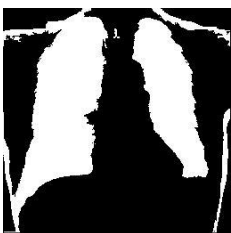


6	06.jpg		256 x 256
7	07.jpg		256 x 256
8	08.jpg		256 x 256
9	09.jpg		256 x 256
10	10.jpg		256 x 256
11	11.jpg		256 x 256

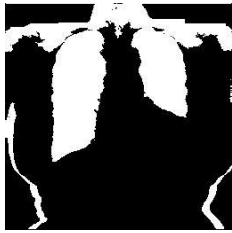

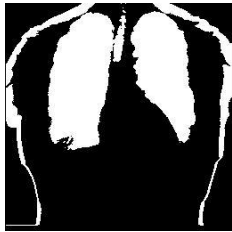
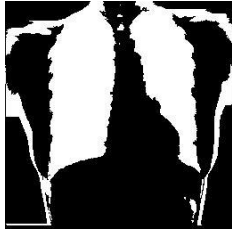
12	12.jpg		256 x 256
13	13.jpg		256 x 256
14	14.jpg		256 x 256
15	15.jpg		256 x 256
16	16.jpg		256 x 256
17	17.jpg		256 x 256

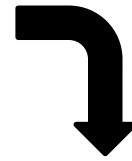
18	18.jpg	 A high-contrast, black and white chest X-ray showing the lungs and ribcage. The image is centered and shows clear lung fields with some rib shadows.	256 x 256
19	19.jpg	 A high-contrast, black and white chest X-ray showing the lungs and ribcage. The image is centered and shows clear lung fields with some rib shadows.	256 x 256
20	20.jpg	 A high-contrast, black and white chest X-ray showing the lungs and ribcage. The image is centered and shows clear lung fields with some rib shadows.	256 x 256
21	21.jpg	 A high-contrast, black and white chest X-ray showing the lungs and ribcage. The image is centered and shows clear lung fields with some rib shadows.	256 x 256
22	22.jpg	 A high-contrast, black and white chest X-ray showing the lungs and ribcage. The image is centered and shows clear lung fields with some rib shadows.	256 x 256
23	23.jpg	 A high-contrast, black and white chest X-ray showing the lungs and ribcage. The image is centered and shows clear lung fields with some rib shadows.	256 x 256

24	24.jpg		256 x 256
25	25.jpg		256 x 256
26	26.jpg		256 x 256
27	27.jpg		256 x 256
28	28.jpg		256 x 256
29	29.jpg		256 x 256

30	30.jpg		256 x 256
31	31.jpg		256 x 256
32	32.jpg		256 x 256
33	33.jpg		256 x 256
34	34.jpg		256 x 256
35	35.jpg		256 x 256

36	36.jpg		256 x 256
37	37.jpg		256 x 256
38	38.jpg		256 x 256
39	39.jpg		256 x 256
40	40.jpg		256 x 256
41	41.jpg		256 x 256

42	42.jpg		256 x 256
43	43.jpg		256 x 256
44	44.jpg		256 x 256
45	45.jpg		256 x 256

4. Source code proses perhitungan algoritme *K-Means*

Deklarasi variabel *struct* sebagai media penyimpanan hasil dari proses perhitungan K-Means:

- Jarak setiap *pixel* terhadap masing-masing *centroid*
- Menyimpan nilai RGB untuk setiap *centroid*

```
public unsafe class KMeans
{
    public class Distance
    {
        public Distance(float d) { _d = d; }
        public float Measure
        {
            get { return _d; }
            set { _d = value; }
        }
        private float _d;
    }

    public class Cluster
    {
        public Cluster(float R, float G, float B)
        {
            _centroid1 = R;
            _centroid2 = G;
            _centroid3 = B;
        }
        public float CentroidR
        {
            get { return _centroid1; }
            set { _centroid1 = value; }
        }
        public float CentroidG
```



```

    {
        get { return _centroid2; }
        set { _centroid2 = value; }
    }
    public float CentroidB
    {
        get { return _centroid3; }
        set { _centroid3 = value; }
    }
    private float _centroid1;
    private float _centroid2;
    private float _centroid3;
}

```

```

public KMeans(Bitmap bmp, int numCluster, Colour.Types
model)
{
    _image = (Bitmap)bmp.Clone();
    _processedImage = (Bitmap)bmp.Clone();
    _model = model;

    _previousCluster = new Dictionary<string, Cluster>();
    _currentCluster = new Dictionary<string, Cluster>();
    FindTopXColours(numCluster); //find top X colours in
the image
    //create clusters for top X colours
    for (int i = 0; i < _topColours.Length; i++)
    {
        PixelData pd =
        Colour.GetPixelData(_topColours[i].R,
        _topColours[i].G, _topColours[i].B, model);

        _previousCluster.Add(_topColours[i].Name, new
Cluster(pd.Ch1, pd.Ch2, pd.Ch3));
        _currentCluster.Add(_topColours[i].Name, new
Cluster(pd.Ch1, pd.Ch2, pd.Ch3));
    }
}

```

Method *KMeans* untuk memulai proses klasterisasi warna citra *x-ray thorax*.

Mengambil dua warna dengan nilai RGB tertinggi dan kemudian dijadikan sebagai sawal.

```

public void Iterate()
{
    _colourClusterAllocation = new
Hashtable(); //for keeping track of
colour<->cluster allocation

```

Method *Iterate* untuk perulangan penentuan *centroid*, dan mengklasterisasi setiap *pixel* warna pada citra.

Memanggil *method*

```

        _pixelDataClusterAllocation = new
Hashtable();
        _clusterColours = new Hashtable();

        UnsafeBitmap fastBitmap = new
UnsafeBitmap(_image);
        fastBitmap.LockBitmap();
        Point size = fastBitmap.Size;
        BGRA* pPixel;

        for (int y = 0; y < size.Y; y++)
        {
            pPixel = fastBitmap[0, y];
            for (int x = 0; x < size.X; x++)
            {
                PixelData pd =
Colour.GetPixelData(pPixel, _model);
                AllocateToCluster(pd);

                //increment the pointer
                pPixel++;
            }
        }
        fastBitmap.UnlockBitmap();
        CalculateClusterCentroids();
        _processedImage = (Bitmap)_image.Clone();

        //segment the image based on the cluster
        fastBitmap = new
UnsafeBitmap(_processedImage);
        fastBitmap.LockBitmap();
        for (int y = 0; y < size.Y; y++)
        {
            pPixel = fastBitmap[0, y];
            for (int x = 0; x < size.X; x++)
            {
                PixelData pd =
Colour.GetPixelData(pPixel, _model);
                Color newClr =
(Color)_clusterColours[pd.Name];

                pPixel->red = newClr.R;
                pPixel->green = newClr.G;
                pPixel->blue = newClr.B;

                //increment the pointer
                pPixel++;
            }
        }
    }

```

```
fastBitmap.UnlockBitmap();  
  
CheckConvergence();  
  
}
```

```

private void CheckConvergence()
{
    //if current and previous cluster centroids are the
    same then converged

    foreach (KeyValuePair<string, Cluster> cluster in
        _currentCluster)

        if (((int)cluster.Value.CentroidR !=
            (int)_previousCluster[cluster.Key].CentroidR)
            && ((int)cluster.Value.CentroidG !=
            (int)_previousCluster[cluster.Key].CentroidG)
            && ((int)cluster.Value.CentroidB !=
            (int)_previousCluster[cluster.Key].CentroidB))
            foreach (KeyValuePair<string, Cluster> cluster in
                _currentCluster)

                _previousCluster[cluster.Key].CentroidR =
                cluster.Value.CentroidR;
                _previousCluster[cluster.Key].CentroidG =
                cluster.Value.CentroidG;
                _previousCluster[cluster.Key].CentroidB =
                cluster.Value.CentroidB;
}

```

**Method**

*CheckConvergence* yang berfungsi untuk melakukan cek status konvergen.

Dengan melakukan perbandingan nilai RGB yang tersimpan pada *previousCluster* dengan *CurrentCluster*.

Variabel *match* menyimpan nilai kondisi *bool* yang berisi *true* atau *false*. Di mana *true* berarti status konvergen telah terpenuhi dan posisi *centroid* tidak berubah.

C

```

private void CalculateClusterCentroids(){
    foreach (KeyValuePair<string, Cluster> cluster in
        _currentCluster){
        List<PixelData> clrList
        (List<PixelData>)_pixelDataClusterAllocation[cluster.Key];
        float cR = 0;
        float cG = 0;
        float cB = 0;
        foreach (PixelData clr in clrList)
        {
            cR += clr.Ch1;
            cG += clr.Ch2;
            cB += clr.Ch3;
            if (!_clusterColours.ContainsKey(clr.Name))
            {
                _clusterColours.Add(clr.Name,
                Color.FromArgb((int)cluster.Value.CentroidR,
                (int)cluster.Value.CentroidG,
                (int)cluster.Value.CentroidB));
            }
        }
        float count = clrList.Count + 1; //total of colours plus 1
        for the existing centroid
        cluster.Value.CentroidR = (cluster.Value.CentroidR + cR) /
        count; //average to find new centroid
        cluster.Value.CentroidG = (cluster.Value.CentroidG + cG) /
        count;
        cluster.Value.CentroidB = (cluster.Value.CentroidB + cB) /
        count;
    }
}
}

```

**Method**

*CalculateClusterCentroid* berfungsi untuk menghitung kembali *centroid* baru.

```

private void AllocateToCluster(PixelData pd)
{
    //find distance of this colour from each cluster centroid
    Dictionary<string, Distance> distances = new
    Dictionary<string, Distance>();

    foreach (KeyValuePair<string, Cluster> c in
    _currentCluster)
    {
        float d = (float)Math.Sqrt(
        (double)Math.Pow((c.Value.CentroidR -
        pd.Ch1), 2) +
        (double)Math.Pow((c.Value.CentroidG -
        pd.Ch2), 2) +
        (double)Math.Pow((c.Value.CentroidB -
        pd.Ch3), 2)
        );
        distances.Add(c.Key, new Distance(d));
    }

    //allocate this colour to the closest cluster based on
    distance
    List<KeyValuePair<string, Distance>> list = new
    List<KeyValuePair<string, Distance>>();
    list.AddRange(distances);

    list.Sort(delegate (KeyValuePair<string, Distance> kvp1,
    KeyValuePair<string, Distance> kvp2)
    { return
    Comparer<float>.Default.Compare(kvp1.Value.Measure,
    kvp2.Value.Measure); });

    //assign to closest cluster
    if
    (_pixelDataClusterAllocation.ContainsKey(list[0].Key))
    {
        ((List<PixelData>)_pixelDataClusterAllocation[list[0].
        Key]).Add(pd);
    }
    else
    {
        List<PixelData> clrList = new
        List<PixelData>();
        clrList.Add(pd);

        _pixelDataClusterAllocation.Add(list[0].Key,
        clrList);
    }
}

```

*Method AllocateToCluster* berfungsi untuk memasukkan setiap *pixel* warna pada citra ke dalam kelompok kluster *centroid* nya. Di mana dalam hal ini hanya terdapat 2 *centroid*.

Perhitungan jarak terhadap 2 *centroid* menggunakan rumus *K-Means*.

```

private void FindTopXColours(int numColours)
{
    Dictionary<string, ColourCount> colours = new
    Dictionary<string, ColourCount>();
    UnsafeBitmap fastBitmap = new
    UnsafeBitmap(_image);
    fastBitmap.LockBitmap();
    Point size = fastBitmap.Size;
    BGRA* pPixel;

    for (int y = 0; y < size.Y; y++)
    {
        pPixel = fastBitmap[0, y];
        for (int x = 0; x < size.X; x++)
        {
            //get the bin index for the current pixel colour
            Color clr = Color.FromArgb(pPixel->red, pPixel->green,
            pPixel->blue);

            if (colours.ContainsKey(clr.Name))
            {
                ((ColourCount)colours[clr.Name]).Count++;
            }
            else
                colours.Add(clr.Name, new
                ColourCount(clr, 1));
            //increment the pointer
            pPixel++;
        }
    }

    fastBitmap.UnlockBitmap();

    //instantiate using actual colours found - which might be
    less than numColours
    if (colours.Count < numColours)
        numColours = colours.Count;

    _topColours = new Color[numColours];
    List<KeyValuePair<string, ColourCount>> summaryList = new
    List<KeyValuePair<string, ColourCount>>();
    summaryList.AddRange(colours);
    summaryList.Sort(delegate (KeyValuePair<string,
    ColourCount> kvp1, KeyValuePair<string, ColourCount> kvp2)
    { return Comparer<int>.Default.Compare(kvp2.Value.Count,
    kvp1.Value.Count); });

    for (int i = 0; i < _topColours.Length; i++)
    {
        _topColours[i] =
        Color.FromArgb(summaryList[i].Value.Colour.R,
        summaryList[i].Value.Colour.G,
        summaryList[i].Value.Colour.B);
    }
}

```

*Method FindTopXColours* berfungsi untuk mencari dua warna dengan nilai RGB tertinggi. Kemudian nilai RGB tertinggi tersebut dijadikan sebagai *centroid*.

Nilai tertinggi RGB dari skala 0-255. Di mana 255 adalah nilai tertinggi.